

Digital design techniques for dependable High-Performance Computing

*Original*

Digital design techniques for dependable High-Performance Computing / Azimi, Sarah. - (2019 May 16), pp. 1-196.

*Availability:*

This version is available at: 11583/2734213 since: 2019-05-27T13:45:42Z

*Publisher:*

Politecnico di Torino

*Published*

DOI:

*Terms of use:*

Altro tipo di accesso

This article is made available under terms and conditions as specified in the corresponding bibliographic description in the repository

*Publisher copyright*

(Article begins on next page)



# ScuDo

Scuola di Dottorato ~ Doctoral School

WHAT YOU ARE, TAKES YOU FAR

Doctoral Dissertation

Doctoral Program in Computer and Control Engineering (31<sup>th</sup> cycle)

# Digital Design Techniques for Dependable High Performance Computing

By

**Sarah Azimi**

\*\*\*\*\*

**Supervisor(s):**

Prof. Luca Sterpone

**Doctoral Examination Committee:**

Prof. Fernanda.Lima Kastensmidt. , Referee, Università federale del Rio Grande do Sul

Prof. Luis.Entrena, Referee, Università Carlos III

Prof. Otmane.Ait Mohamed, University of Concordia

Prof. Monica.Alderighi, Istituto di Astrofisica Spaziale

Prof. Stefano.Di Carlo, Politecnico di Torino

Politecnico di Torino

2019

## Declaration

I hereby declare that, the contents and organization of this dissertation constitute my own original work and does not compromise in any way the rights of third parties, including those relating to the security of personal data.

Sarah Azimi  
2019

\* This dissertation is presented in partial fulfillment of the requirements for **Ph.D. degree** in the Graduate School of Politecnico di Torino (ScuDo).

*To my Mum and Dad, the hidden strength behind my every success.*

## **Acknowledgements**

I would like to acknowledge all the people who support me and encourage me to accomplish my Ph.D thesis.

Many thanks goes to Prof. Luca Sterpone for providing me the opportunity to pursue my research activity in the CAD group and for his great support and generous help over the three pleasant years of my Ph.D. I would like to thank Dr. Boyang Du for his fundamental support during my research activity. I also would like to thank all the people I had the chance to have the collaboration with, specially David Merodio Codinacs From European Space Agency for all his support during my Ph.D. I also acknowledge Prof. Matteo Sonza for his fruitful encouragement during my Ph.D program. I would like to thank all my colleague in the CAD group and Lab3 for all the pleasant time we spent together.

Special thanks goes to Mariangela Saracco for her great support and effort in all the steps.

## Acknowledgements

- To my Mum and Dad who scarify their lives to give me a better life.
- To Luca, my best friend, who taught me about dreams and how to catch them.
- To Samira, my sister, who taught me that the light is always there.
- **To Hannah, whose brought joy and hope to our lives.**
- To Boyang who showed me how it feels to have the support of a brother.
- To Antonio who reminds me how lucky I am.
- To CAD & Reliability Group which provides me the opportunity to grow and build a bright future.
- To Lab3 which ties my life with amazing people.

## **Abstract**

The overall goal of my doctoral research activity is oriented in development of high performance computing design techniques for high reliability of digital circuits. One of the most critical environment aspect that could reduce the reliability of modern VLSI technologies used for high performance computing is radiation. When a set of radiation particle interact within the electronic systems by an exchange of energy, several kinds of effects can be observed. The impact of radiation effects on electronic devices can cause a misbehavior on the functionality of the circuit. In order to apply strategies and techniques to tolerate these faults and errors, these effects must be analyzed in details. Considering high frequency and smaller size of recent technologies, the sensitivity of High Performance Computing toward radiation is expected to be higher. Therefore, having more resilient mitigation technologies is more relevant and necessary, which is the focus of my research activity.

Radiation-induced effects can lead different effects depending on the location and time of the incident. If the effects of radiation incident lasts for a short period of time, it is known as a transient fault. While, if the effects last for a longer duration, it is known as a permanent fault. Therefore, my PhD dissertation is divided into two main parts. The first part as the main part is dedicated to the transient fault, mostly focusing on Single Event Transient while the second part is dedicated to the more permanent fault such as Micro Single Event Latch-up and Total Ionizing Dose.

Considering Single Event Transient as the golden part of my research, it covers different phases of this phenomena from the generation until the mitigation. The first phase is dedicated to the physical modeling of these effects and evaluating the impact of the radiation environment profile on the generated SET pulse. The second phase is devoted to develop tools and algorithms for analyzing and predicting the behavior of the effected device. As a last phase, the developed physical model and performed analysis have been the golden keys to propose an efficient mitigation solution for

robustness of the developed system against this phenomena. The proposed mitigation solution has been known as the first method able to filter Single Event Transient pulses with zero-timing overhead. These methodologies have been applied to modern High Performance Computing technologies with high frequency and smaller size which lead to more critical condition for Single Event Transient effect.

This comprehensive proposed flow for analyzing and mitigating Single Event Transient has been applied to several industrial projects such as EUCLID space mission project with the goal of monitoring the dark space which the launch planned for 2020 carrying by European Space Agency. The developed SET analysis and mitigation work-flow has been part of the handbook *Space Product Assurance Techniques for Radiation Effects Mitigation in ASICs and FPGAs handbook*, published by European Space Agency. Moreover, the developed set of tools has been known as the *Best EDA Tool for improving design automation for integrated circuits and systems* by IEEE Council on Electronic Design Automation.

However, as a part of my research activity, not only I focused on the Transient effects, but I dedicated the second part of my dissertation to the evaluation of permanent effects such as Single Event Latch-up and total Ionizing Dose. The second effect I focused on is Single Event Latch-up which is one of the major reliability concerns for VLSI device applied in safety critical applications. The reduction of the circuit feature size and operating voltage levels are leading to a new kind of latch-up called micro Single Event Latch-up. Single Event Latch-up tends to occur near the input/output terminals of logic gates while micro Single Event Latch-up may occur at various locations between layers. One of my main research contribution is to propose a first 3D model for describing the 3D physical layout description of the design including the interconnection resources and logic versatile. This 3D layout description leads to analyze the sensitivity of the sub-micron circuitry to Micro Single Event Latch-up phenomena with respect to the layout, depth, size and density of the design. This methodology is considered as the first one applicable to large industrial designs.

Bench-marking technologies are becoming increasingly attractive since their configuration memory is almost immune to Single Event Upset. However, applied in mission critical application, especially long term missions, the FPGA devices are subject to cumulative ionizing damage, known as Total Ionizing Dose. Total Ionizing Dose may affect the FPGA, causing performance degradation and eventually



permanent damage. Therefore, I dedicated part of my research activity to propose a physical model of Total Ionizing Dose effect in order to analyze the Total Ionizing Dose effect on recent modern technologies.

# Table of contents

<b>List of Figures</b>	<b>xiii</b>
<b>List of Tables</b>	<b>xx</b>
<b>1 Introduction on Radiation Effect on Modern VLSI Technologies</b>	<b>1</b>
1.1 Radiation characteristics . . . . .	3
1.2 Modern VLSI Technologies . . . . .	6
1.2.1 Field Programmable Gate Array . . . . .	6
1.2.2 GPGPU . . . . .	9
<b>I Single Event Transient</b>	<b>12</b>
<b>2 Basic Mechanism of Single Event Transient</b>	<b>14</b>
2.1 From radiation particle to voltage pulse . . . . .	14
2.1.1 Basic mechanism of Single Event Transient . . . . .	15
2.1.2 SET life-cycle inside the device . . . . .	16
<b>3 Analysis of Single Event Transient</b>	<b>18</b>
3.1 SET Characterization . . . . .	18
3.1.1 Electrical SET Injection . . . . .	19
3.1.2 SET propagation characterization . . . . .	20

3.1.3	SET characterization test setup on Flash-based FPGAs . . .	22
3.1.4	SET characterization test setup on SRAM-based FPGAs . .	31
3.1.5	Research advancements on SET characterization . . . . .	40
3.2	On the prediction of SETs . . . . .	40
3.2.1	SET physical dynamic simulation model . . . . .	41
3.2.2	SET prediction methodology . . . . .	45
3.2.3	Prediction of SET on Flash-based FPGAs . . . . .	47
3.2.4	Research advancement on the prediction of SETs . . . . .	48
3.3	Single Event Transient Analyzer - SETA . . . . .	50
3.3.1	SET behavior in combinational logic . . . . .	51
3.3.2	SET behavior in routing interconnections . . . . .	52
3.3.3	Integration of SETA with commercial tools . . . . .	53
3.3.4	SETA on Flash-based FPGAs . . . . .	57
3.4	Evaluation of Transient Errors in GPGPUs . . . . .	59
3.4.1	The proposed environment . . . . .	61
3.4.2	Fault tolerance design methods on GPGPU . . . . .	67
3.4.3	Experimental results . . . . .	77
3.5	Convergence Single Event Transient Analyzer - CSETA . . . . .	84
3.5.1	SET pulse behavior at convergence point . . . . .	84
3.5.2	Integration of CSETA with commercial tools . . . . .	87
3.5.3	C-SETA on Flash-based FPGAs . . . . .	90
3.5.4	Research advancement on Single Event Transient Analyzer .	93
<b>4</b>	<b>Mitigation of Single Event Transient</b>	<b>94</b>
4.1	Guard Gate mitigation . . . . .	95
4.1.1	SET propagation analysis . . . . .	96
4.1.2	Netlist filter mitigation insertion . . . . .	97

4.1.3	Physical implementation . . . . .	99
4.1.4	Guard Gate Mitigation on Rad-Hard RTG4 Flash-based FPGAs	101
4.2	SET Mitigation by adding Charge Sharing logics on Flash-based FPGA . . . . .	104
4.2.1	Proposed design flow . . . . .	106
4.2.2	Experimental results . . . . .	109
4.2.3	Research advancement on mitigation of Single Event Transient	112
<b>5</b>	<b>Industrial Application</b>	<b>114</b>
5.1	Radiation Test: Ultra High Energy Heavy Ion Test Beam on Xilinx Kintex-7 SRAM-based FPGA . . . . .	114
5.2	Background . . . . .	115
5.3	Device and Design Under the Test . . . . .	117
5.4	Monitoring setup . . . . .	120
5.5	UHE Heavy Ion beam . . . . .	120
5.6	Radiation Test Data . . . . .	122
5.6.1	Error rate analysis . . . . .	122
5.6.2	Observation of SEMU . . . . .	124
5.7	EUCLID Space Mission . . . . .	128
5.7.1	What is EUCLID? . . . . .	128
5.7.2	Analysis of EUCLID original EUCLID netlist sensitivity to SET . . . . .	130
5.7.3	Mitigating the EUCLID design netlist . . . . .	132
5.7.4	Analysis of EUCLID mitigated netlist sensitivity to SET phenomena . . . . .	134

<b>II</b>	<b>From Transient to Permanent</b>	<b>137</b>
<b>6</b>	<b>Micro Single Event Latch-up</b>	<b>139</b>
6.1	From SEL to Micro SEL . . . . .	140
6.1.1	Micro Single Event Latch-up Analysis . . . . .	142
6.1.2	Experimental Results . . . . .	147
6.1.3	Research advancement on Micro Single Event Latch-up . . .	150
<b>7</b>	<b>Total Ionizing Does</b>	<b>152</b>
7.1	The developed environment . . . . .	153
7.1.1	Background on Versatile architecture . . . . .	154
7.1.2	Total Ionizing Dose (TID) Heatmap generation . . . . .	155
7.1.3	Hitlist generation . . . . .	157
7.1.4	SDF Instrumentation . . . . .	159
7.1.5	Simulation Execution . . . . .	159
7.2	Experimental Results . . . . .	160
7.2.1	Experimental Setup . . . . .	162
7.2.2	Error Rate Reports . . . . .	164
7.2.3	Research advancement on Total Ionizing Does . . . . .	165
	<b>References</b>	<b>167</b>
	<b>Appendix A Research Achievements</b>	<b>174</b>

# List of Figures

1.1	FPGA General Architecture . . . . .	7
1.2	Logic Block (SLICEL) diagram from Virtex-5 device of Xilinx . . .	8
1.3	A routed design mapped on ProASIC3 from Microsemi . . . . .	9
1.4	Contemporary GPU architecture . . . . .	10
2.1	Floating gate transistor layout in the 130 nm Flash-based FPGA and the correspondent sensitive node generating transient pulses . . . .	15
2.2	Examples of transient pulses generated by our electrical fault injection platform and mimic heavy ions radiation particles hitting the sensitive nodes of a 130nm Flash-based FPGAs . . . . .	16
2.3	Propagation of SET pulse through an Inverter with an input transition of 0-1-0 . . . . .	17
3.1	The developed logic scheme of internal injection generation . . . .	20
3.2	A Logic sensitive node and the two observability methods: towards next logic gates and toward fan-out . . . . .	21
3.3	Scheme of internal electrical pulse generator . . . . .	22
3.4	Logical scheme of the test- scenario 1 . . . . .	24
3.5	An overview of the placement layout- Scenario 1 . . . . .	25
3.6	Propagation Induced Pulse Broadening- Scenario 1 (Source SET = 3.5 ns) . . . . .	25
3.7	Inverter chain delay- Scenario 1 . . . . .	26

3.8	Logical scheme of the test- Scenario 2 . . . . .	26
3.9	An overview of the placement layout- Scenario 2 . . . . .	27
3.10	Propagation Induced Pulse Broadening(PIPB)- Scenario 2 (Source SET = 3.5 ns) . . . . .	27
3.11	Inverter chain delay- Scenario 2 . . . . .	28
3.12	Logical scheme of full adder by SET in the divergence point . . . . .	28
3.13	Conceptual scheme- third scenario . . . . .	29
3.14	PIPB report- third scenario (Source SET = 3.5 ns) . . . . .	29
3.15	Delay report- third scenario . . . . .	30
3.16	Logical scheme of full adder affected by SET in the convergence point . . . . .	30
3.17	Conceptual scheme- fourth scenario . . . . .	31
3.18	PIPB report- fourth scenario (Source SET = 3.5 ns) . . . . .	31
3.19	Delay report- fourth scenario . . . . .	32
3.20	An example of SET propagated through two convergence paths and generating: two independent SET pulse (a) and C-SET pulse (b) . . . . .	32
3.21	Overview of the global analysis methodology for SRAM-based FPGAs . . . . .	34
3.22	Scheme of the SET pulse measurement circuit . . . . .	35
3.23	Simulation results for injecting SET of 0.2 ns to a chain of 100 INVs . . . . .	37
3.24	Propagation pulse broadening for inverting gates- each chain includes 100 gates . . . . .	38
3.25	Propagation pulse broadening for non-inverting gates . . . . .	38
3.26	Propagation pulse broadening for circuit benchmarks implemented on SRAM-based FPGAs . . . . .	40
3.27	An example of dynamic LET form on a generic empty layout in 2D and 3D . . . . .	42
3.28	The flow of the developed SET prediction method . . . . .	45
3.29	The flow of the developed SET prediction method . . . . .	46
3.30	Global scheme of the generated test setup. . . . .	47

3.31	Scheme of the SET pulse measurement circuit. . . . .	48
3.32	Comparison between the SET prediction model and the heavy-ion test results. Standard error deviation at 10%. . . . .	49
3.33	Comparison between the SET prediction model and the heavy-ion test results. Standard error deviation at 1%. . . . .	49
3.34	Comparison between the SET prediction model and the heavy-ion test results. Standard error deviation at 0.1%. . . . .	50
3.35	The device routing topology of the Microsemi ProASIC3 family. . .	52
3.36	The developed analysis flow for the accurate evaluation of SET effects on SoC implemented on Flash-based FPGA. . . . .	54
3.37	The SET propagation approach: a voltage vector $PG_i$ is transformed in a new vector $PG_{i+1}$ , considering the gate $G_i$ , the subsequent gate $G_{i+1}$ and the routing segment between two logic gates. . . . .	55
3.38	The main core of the Propagation Induced Pulse Broadening (PIPB) calculation for the involved gates gate $G_i$ and the following gate $G_{i+1}$ (red X is representing the worst condition while green x is representing the best condition). . . . .	56
3.39	Representation of the cumulative (KPIPB) on a generic couple of gates considered in a pulse traversing computation (red X is representing the worst condition while green x is representing the best condition). . . . .	57
3.40	The flow of the developed simulation-based fault injection for transient errors analysis on GPGPUs. . . . .	62
3.41	GPGPU-sim modeled system. . . . .	64
3.42	Soft-error injection tool integration in the GPGPU-sim simulator. . .	65
3.43	Soft-error injection tool integration in the GPGPU-sim simulator. . .	66
3.44	The matrix multiplication kernel algorithm. . . . .	69
3.45	Matrix product implementation comparison with and without using the shared memory. The developed algorithm is using the shared memory thus getting an improvement in performances. . . . .	70



3.46	The algorithm of matrix multiplication with DWC. . . . .	71
3.47	The algorithm of Matrix Multiplication with TMR-kernel. . . . .	72
3.48	Additional buffers scheme needed for the check-sums for the matrix product ABFT method. . . . .	72
3.49	Scheme of the shuffle operation on the FF network to the auxiliary network done in the global memory buffers . . . . .	73
3.50	FFT host algorithm executed on the share memory . . . . .	73
3.51	FFT propagation kernel algorithm executed on the shared memory .	74
3.52	The comparison of the FFT implementation with and without using the shared memory. . . . .	74
3.53	The check-pointing scheme adopted integrating the ABFT algorithm.	75
3.54	The FFT algorithm with the ABFT mean based check pointing. . . .	75
3.55	Comparison of the execution times of the Sobel operator implemented in Frequency (blue line) and in space (red line). . . . .	76
3.56	An example of loical cone outputs. . . . .	78
3.57	Single streaming processor SET sensitivity overview for injecting 1000 SET pulses. . . . .	79
3.58	An example of SET encountering the divergence point and convergence point. . . . .	85
3.59	Outcome of SET at the convergence point- C-SET. . . . .	86
3.60	Correlation between maximum width of C-SET and Difference of delays between two paths. . . . .	86
3.61	Scheme of developed flow for accurate analysis of C-SET. . . . .	88
3.62	The pseudo code for the identification of the SET width and amplitude in a post layout circuit. . . . .	89
3.63	Classification of C-SET in terms of criticality. . . . .	92
4.1	The Overview of the SET-aware mitigation flow including the SET propagation analysis, the Netlist filter mitigation insertion and the marco-oriented mapping and filtering-driven place and route. . . . .	96

4.2	The SET propagation method including the PIPB computation on the propagation node. . . . .	97
4.3	Traditional Flip-Flop based guard-gate and SET-Filtering solution(a)compared to the SET-filtering scheme inserted by the netlist insertion mapper on logic gates shared by logic cones(b). . . . .	98
4.4	An example of the netlist mitigation algorithm as a first phase on a circuit portion: the calculation of the broadening coefficients in nanoseconds. . . . .	99
4.5	Application of the netlist mitigation algorithm second phase to a circuit portion: the insertion of the filtering scheme on the shared logic path. . . . .	100
4.6	The global placement implementation algorithm. . . . .	101
4.7	The functional block diagram of logic element of the RTG4 Flash-based FPGA family. . . . .	102
4.8	Overview of the developed flow including SET analysis and charge sharing mitigation. . . . .	106
4.9	The charge sharing mitigation algorithm for Flash-based FPGAs. . .	107
4.10	Charge sharing number of gates per logic nodes with respect to the routing delay and PIPB coefficient. . . . .	109
4.11	The key concept of the Charge Sharing mitigation algorithm. . . . .	110
5.1	SEU in configuration memory may corrupt circuit design mapped on FPGA . . . . .	115
5.2	SEUs in configuration memory affects different copies of logic path in XTMR implementation . . . . .	116
5.3	Original ARM-based SoC used as benchmark circuit . . . . .	118
5.4	Replication scheme of ARM-based SoC for increasing device utilization . . . . .	119
5.5	Monitor flow with the Host PC application . . . . .	121
5.6	board and beam setup for alignment. . . . .	121

5.7	VERI_Place error rate comparison with radiation test data for plain version . . . . .	122
5.8	VERI_Place error rate comparison with radiation test data for XTMR version . . . . .	123
5.9	VERI_Place error rate comparison between the Plain and XTMR version collected during radiation test . . . . .	123
5.10	Application and configuration memory error-rate cross-section comparison for Plain and XTMR versions. . . . .	124
5.11	Cluster (SEMU) patterns observed during radiation test. . . . .	126
5.12	Cluster distribution cross-section of different cluster sizes. . . . .	126
5.13	The EDA adapted flow integrates both commercial tool (Micossemi Libero Soc 11.8) and the SET analysis and mitigation flow. . . . .	130
5.14	The SET distribution obtained on the original EUCLID netlist. . . . .	132
5.15	An example of Gaurd-Gate automation insertion on a portion of the EUCLID design. . . . .	133
5.16	The SET distribution obtained on the Mitigated EUCLID netlist. . . . .	135
5.17	A comparison of SET distribution between the original netlist and the mitigated netlist. . . . .	136
6.1	Electrical effect generating a SEL effect . . . . .	141
6.2	Overview of the basic mechanisms generating a micro SEL effect on the output of a gate . . . . .	141
6.3	The intra-metal layer micro-SEL effect between routing segment. The evidenced red routes represents the affected net . . . . .	142
6.4	OVERview of the global analysis methodology for micro-latch up consisting on the layer mesh-map and the micro-SEL insertion tool . . . . .	143
6.5	A Pseudo-code overview of the developed latch-up analysis environment . . . . .	145
6.6	The flow of the developed Monte Carlo Error Rate Analysis . . . . .	145

6.7	Mutual layer distribution in terms of are width and length for bench- mark B14 version F . . . . .	148
6.8	Metal Layers 1 and 2 for the B14 implementation with routing congestion F(a) and routing congestion A(b) . . . . .	149
6.9	Monte Carlo fault report for the different B14 Physical placement and routing with statistical error rate bar at 1% . . . . .	149
6.10	$\mu$ SEL fault simulation instrumentation method . . . . .	150
7.1	The flow of the developed TID analysis environment . . . . .	154
7.2	VersaTile in Microsemi ProASIC Flash-based FPGA [1] . . . . .	155
7.3	Heatmap generation considering TID distribution: The area A repre- sent a high density TID region, while area B is representing a TID not affecting region. . . . .	156
7.4	Block diagram of TIC Heatmap generation . . . . .	157
7.5	Algorithm for generation Hitlist . . . . .	158
7.6	Algorithm for generation of instrumented SDF. . . . .	159
7.7	Performance degradation for different types of gates [2] . . . . .	163
7.8	Net propagation delay with respect to Manhattan distance between net source and destination(X distance and Y distance separated) . .	163
7.9	Performance degradation coefficient model for routing net . . . . .	164
7.10	Error rate results of the selected ITC99 bench-mark with respect to different percentage for TID . . . . .	165

# List of Tables

3.1	Generated SET using internal electrical injection. . . . .	37
3.2	Benchmark area utilization. . . . .	39
3.3	Benchmark critical path resources. . . . .	39
3.4	$t_{pHL}$ and $t_{pLH}$ for each Inverter . . . . .	44
3.5	Propagation behavior . . . . .	44
3.6	Routing topology organization on ProASIC3 devices . . . . .	53
3.7	Characteristic of the original benchmark circuits . . . . .	57
3.8	Sensitivity report of the selected circuits for injection of 5000 SET pulses lower than 1 ns . . . . .	58
3.9	Matrix multiplication application results fault injection results . . . .	81
3.10	Fast Fourier transform application fault injection results. . . . .	82
3.11	Fast Fourier transform application fault injection results. . . . .	83
3.12	Characteristics of the original benchmark circuits . . . . .	91
3.13	Comprehensive SET sensitivity using static analysis tool- 5000 SET pulses lower than 1 ns are injected. . . . .	92
4.1	Characteristics of the implemented circuits . . . . .	103
4.2	SET fault simulation results . . . . .	104
4.3	Characteristics of the original benchmark circuits . . . . .	109
4.4	Comprehensive Flip-Flop SET sensitivity using the static analysis tool111	

---

4.5	SET fault injection wrong answers comparison . . . . .	112
4.6	Timing and area overhead for each method . . . . .	112
5.1	Resource utilization for Plain and XTMR Version of ARM-based SoC on Kintex-7 . . . . .	119
5.2	Comparison with Test Result of Lower Energy Beam on Kintex-7 . .	127
5.3	Circuit Resources of the EUCLID netlist by type . . . . .	130
5.4	Timing resources of the EUCLID netlist . . . . .	131
5.5	Single Event Transient Analysis for SET Ranging from 0.43 sn to 0.52 ns representing the number of Flip-Flops for each case . . . .	131
5.6	Timing analysis for three iteration of Guard-Gate mitigation tool . .	134
5.7	Area Over-head Report for Three Iteration of Guard Gate Mitigation Tool . . . . .	134
5.8	Circuit Resources for the Mitigation Netlist . . . . .	134
6.1	Routing Architecture Characteristic . . . . .	148
6.2	Dynamic error rate report . . . . .	150
7.1	Benchmark circuits characteristics . . . . .	161

# **Chapter 1**

## **Introduction on Radiation Effect on Modern VLSI Technologies**

Nowadays, electronic devices are used in a growing number of applications, starting from personal computers and entertainment market to large-scale business frameworks such as automobiles and satellites. Each application has its own requirements. However, they are considered as mission critical applications if they are involved in a huge amount of money or human lives. When electronic devices, in particular digital circuits are used in mission critical applications, dependability of these devices is becoming an important issue. Dependability can be defined as the potential to tolerate faults happening due to environmental features, leading to the possible failure of the entire system. Misbehavior of the internal component of a system which is known as fault can propagate until the output of the system and become an error. Finally, if the generated error produced a misbehavior in the functionality of the system, the system is facing a failure [3]. Therefore, strategies and techniques are needed to tolerate faults and errors. In order to provide the most sufficient fault tolerant techniques, faults and errors themselves need to be analyzed and studied in details. These faults and errors can be introduced both from the user side or the surrounding environment. One of the main environment aspects that can reduce the reliability of modern technologies specially when they are used in mission critical application is radiation.

When a set of radiation particles interact within the electronic systems by an exchange of energy, several kinds of effects can be observed. The impact of radiation

effects on electronic devices can cause a misbehavior on the functionality of the circuit. In order to apply strategies and techniques to tolerate these faults and errors, these effects must be analyzed in detail. Considering high frequency and smaller size of recent technologies, the sensitivity of High Performance Computing toward radiation is expected to be higher. Therefore, having more resilient mitigation technologies is more relevant and necessary, which is the focus of my research activity.

Radiation-induced effects can lead different effects depending on the location and time of the incident. If the effects of radiation incident last for a short period of time, it is known as a transient fault. While if the effects last for a longer duration, it is known as a permanent fault. Therefore, my PhD dissertation is divided into two main parts. The first part as the main part is dedicated to the transient fault, mostly focusing on Single Event Transient while the second part is dedicated to the permanent fault such as Micro Single Event Latch-up and Total Ionizing Dose. Considering Single Event Transient as the golden part of this dissertation, my research covers different phases of this phenomenon from the generation until the mitigation. Chapter 1 of this dissertation is dedicated to the elaboration of radiation effect on the modern technologies while chapter 2 is dedicated to the basic mechanism of Single Event Transient, evaluating the SET life-cycle inside the device. The thesis continues in chapter 3 by developing environments and setups for performing characterizations of SET pulses in different devices, development of tools and algorithms for analyzing and predicting the behavior of SET in the effected device. The performed analysis have been the golden keys to propose an efficient mitigation solution for robustness of the developed system against this phenomenon elaborated in chapter 4. The proposed mitigation solution has been known as the first method able to filter Single Event Transient pulses with zero-timing overhead. These methodologies have been applied to modern High Performance Computing technologies with high frequency and smaller size which leads to more critical condition for Single Event Transient effect.

This comprehensive proposed flow for analyzing and mitigating Single Event Transient has been applied to several industrial projects such as EUCLID space mission project with the goal of monitoring the dark space with the lunch planned for 2020 carrying by European Space Agency which is the focus of chapter 5. The developed SET analysis and mitigation work-flow has been part of the handbook *Space Product Assurance Techniques for Radiation Effects Mitigation in ASICs and*



*FPGAs handbook*, published by *European Space Agency*. Moreover, the developed set of tools has been known as the *Best EDA Tool for improving design automation for integrated circuits and systems* by *IEEE Council on Electronic Design Automation*.

However, as a part of my research activity, not only I focused on the Transient effects, but I dedicated the second part of my dissertation to the evaluation of permanent effects such as Single Event Latch-up and total Ionizing Dose. Chapter 6 focuses on Single Event Latch-up which is one of the major reliability concerns for VLSI device applied in safety critical applications. The reduction of the circuit feature size and operating voltage levels are leading to a new kind of latch-up called micro Single Event Latch-up. Single Event Latch-up tends to occur near the input/output terminals of logic gates while micro Single Event Latch-up may occur at various locations between layers. One of the main research contributions of this dissertation is to propose a first 3D model for describing the 3D physical layout description of the design including the interconnection resources and logic versatile. This 3D layout description leads to analyze the sensitivity of the sub-micron circuitry to Micro Single Event Latch-up phenomena while considering the layout, depth, size and density of the design. This methodology is considered as the first one applicable to large industrial designs.

Bench-marking technologies are becoming increasingly attractive since their configuration memory is almost immune to Single Event Upset. However, applied in mission critical application, especially long-term missions, the FPGA devices are subject to cumulative ionizing damage, known as Total Ionizing Dose. Total Ionizing Dose may affect the FPGA, causing performance degradation and eventually permanent damage. Therefore, chapter 7 is dedicated to propose a physical model of Total Ionizing Dose effect in order to analyze the TID effect on recent modern technologies.

## 1.1 Radiation characteristics

Radiation can be defined as a set of particles that are interacting within the device, transferring their energy to the device and creating several effects. In space, there are several kinds of radiation particles that can easily move in the space environment and interact within the electronic devices, such as: energetic electrons, protons, alpha particles, and heavy-ion particles [4] [5]. Some of these particles such as heavy ions

have a really high energy which are able to overcome the package protection of the chip and produce faults. While, other particles such as alfa particles are with lower energy which reduce the probability of passing through the shielding of the device. However, if these particles are generated inside the device as an interaction of high energy particles with the silicon of the device, they are capable of producing faults.

When an electronic device is exposed to radiation, several kinds of effects can be observed which is dependent to several factors. First of all, generated effects depends on the feature of radiation environment and particles such as the energy of the particle and the incident angel. On the other side, the triggered effects depends on the material of the device that the particle is going though. Considering different material, Linear Energy Transfer (LET) of the particle which is defined as the energy of the particles transferred to the material is different. The LET value plays an important roles in the computation of error rates for electronic system components. In the studies of radiation effects on electronic devices, LET is usually expressed in units of  $MeVcm^2/mg$  of the material, typically silicon, which is representing the lost energy of the particle to the material per unit path length ( $MeVcm^2/cm$ ) divided by the density of the material  $mg/cm^3$ .

Radiation induced faults can be classifies into two main groups: Single Event Effects (SEEs) and Total Ionizing Does (TID). Single Event Effects are happening due to the single particle strike in a certain location of the device. Based on the incident location, electrical field and the energy of the incident particle, different faulty behavior is expected. Theses faulty behavior could be temporary which means that it is going to disappear after a while. These errors are known as Soft Errors. On the other hand, if the faulty behavior is permanent by damaging the device itself, it is known as Hard Errors. Considering the complexity of new technology and shrinking of the device size, Single Event Transient as one of the main soft errors, has been considered as one of the main critical soft errors while Single Event Latch-up is known as one of the main recent hard errors.

Regarding Single Event Transient, when an electronic device is exposed to radiation, the highly charged particles hit the device and interact within the material of the device. The interaction of the ion with the pn junction of the active area of the transistor can break temporally the barrier of the junction provoking a transient pulse of the voltage in that junction which is called Single Event Transient (SET). The SET shape depends not only on the incident but also on the device it strikes. In

fact, it is a function of the LET of the particle and its incident angle, the material encounter in its path inside the device and the electrical fields present at the particular moment [6]. SETs are transient faults with the duration between picoseconds and nanoseconds within the circuit, depending on the pulse width and amplitude. the SET pulse can be generated inside a memory cell and if it presents enough amplitude and duration, it can provoke a bit-flip in the memory cell. Also, SET can happen in transistor of a combinational logic gate cell and be propagated by the circuit until be captured by a memory cell or register. In both cases, bit-flip or SET capture, the effect is an error in the circuit that may be a failure in the future if not masked and leading to misbehavior of the system. Considering the high frequency of high performance computing system, the probability of SET pulses being sampled is drastically high which leads SETs to be a critical phenomenon for modern high performance computing systems.

The effect of radiation particle interaction within the device is not always transient. In fact, some effects are considered as permanent errors that are eliminating just by restarting the device. As a sample of this kind of effects, Single Event Latch-up (SEL) can be mentioned. SEL is happening due to the increasing of the device current as a result of radiation incident. This phenomena usually leads to the destruction of the device itself if not removed in time. The unique method to remove SELs is powering off the device. Recently, due to the reduction of circuit feature size and operating voltage level, a new kind of latch-up event called micro latch-up or micro Single Event Latch-up is observed [7]. Usually, Single Event Latch-up tends to occur near the input/output terminals of a logic gate where a charged particle can pass through the silicon device region as a permanent effect. On the other side, micro latch-up may occur at various locations across the die and between layers, temporarily affecting the logical behavior of technology cells at various location across the die provoking circuit misbehavior. Considering recent Ultra-scale devices based on technologies below 20nm which are well known due to their elevated computing features and low power consumption, these devices are more vulnerable for Micro-latch up effect.

Different from the Single Event Effects, total ionizing dose (TID) is the effect of the accumulation of the charge injected by radiation. The accumulated charge is a function of exposure time, the flux of the particles and the linear energy transfer (LET) of the particles. TID causes three kinds of effects: performance degradation, increase of the power consumption and programmability loss. Performance degradation leads

to a slower device with reduction of maximum frequency while the increase of power consumption leads to higher leakage current, increasing the power consumption when transistors are not used. Eventually, the third effects leads to losing the reprogrammability in the FPGA configuration memory.

Considering the critical affect of theses phenomenon, I dedicated my Ph.D research activity to evaluate the effects of these phenomena to the functionality of the system and providing methodologies to robust the circuit against these effects.

## **1.2 Modern VLSI Technologies**

### **1.2.1 Field Programmable Gate Array**

Field Programmable Gate Arrays (FPGAs) are becoming more and more commonly used in various application fields due to their flexibility and low development cost. Furthermore, with technology scaling, the computing power they can provide keeps increasing while the cost and power consumption remains low. This makes them even attractive in safety- and mission-critical fields such as automotive, avionics and space applications.

The FPGA, by different manufacture process technology, can be divided into SRAM-based FPGA, Flash-based FPGA and so on (Antifuse, EEPROM etc). SRAM-based FPGA, such as the ones from Xilinx and Altera, provide large amount of on-chip resources including Logic Blocks, Digital Signal Processing (DSP) Units, On-chip Memory etc. Together with high performance, low power consumption and high flexibility via partial reconfiguration, these features make SRAM-based FPGA very popular in the market. While comparing to SRAM-based FPGA, Flash-based FPGA does not require extra memory device to store configuration file, and does not require to reprogram after each power-on, due to the non-volatile Flash-based configuration memory. More importantly, the configuration memory inside Flash-based FPGA are almost immune to permanent loss of configuration data. Thus, Flash-based FPGA are gaining more and more interest in space and avionic applications.

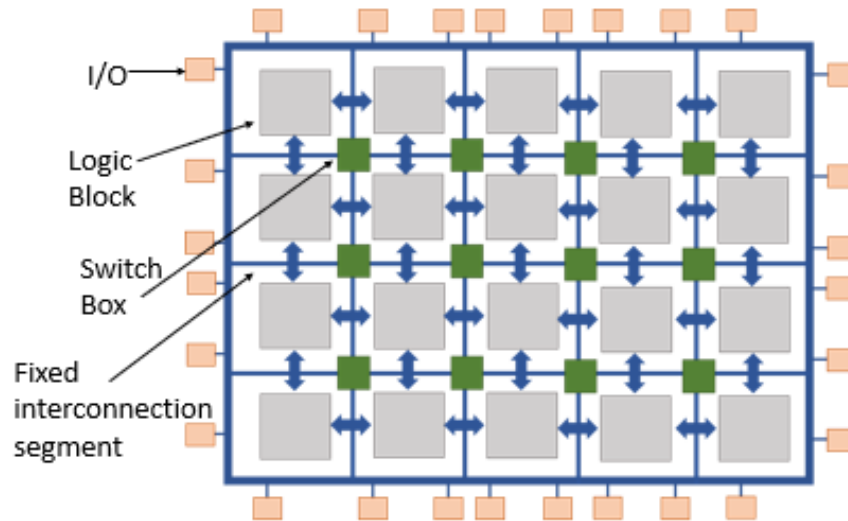


Fig. 1.1 FPGA General Architecture

Though internal architectures may vary from device to device, the SRAM-based FPGA and Flash-based FPGA share the same general architecture as shown in Figure 1.1 which includes Logic Blocks and Switch Boxes.

- The Logic Blocks typically contains resources such as Look Up Table (LUT), Multiplexer and Registers that user can configure according to the logic circuit to be implemented and mapped on the FPGA. An example of Logic Block (from Xilinx Virtex-5) is shown in Figure 1.2.
- The Switch Boxes usually contains interconnection segments which can be configured as active or inactive as required by the routing of the implemented circuit on FPGA. As example of routed design mapped on FPGA (ProASIC3 from Microsemi) is shown in Figure 1.3.

With other resources provided such as Block Memories, PLLs etc., designer can create complicated system on FPGA using high level Hardware Description Language (HDL) or High Level Synthesis methods supported by vendor tools. A complete design flow usually includes several steps. The synthesis tool compiles the design files in HDL or other high level design format to Gate-Level netlist, and then

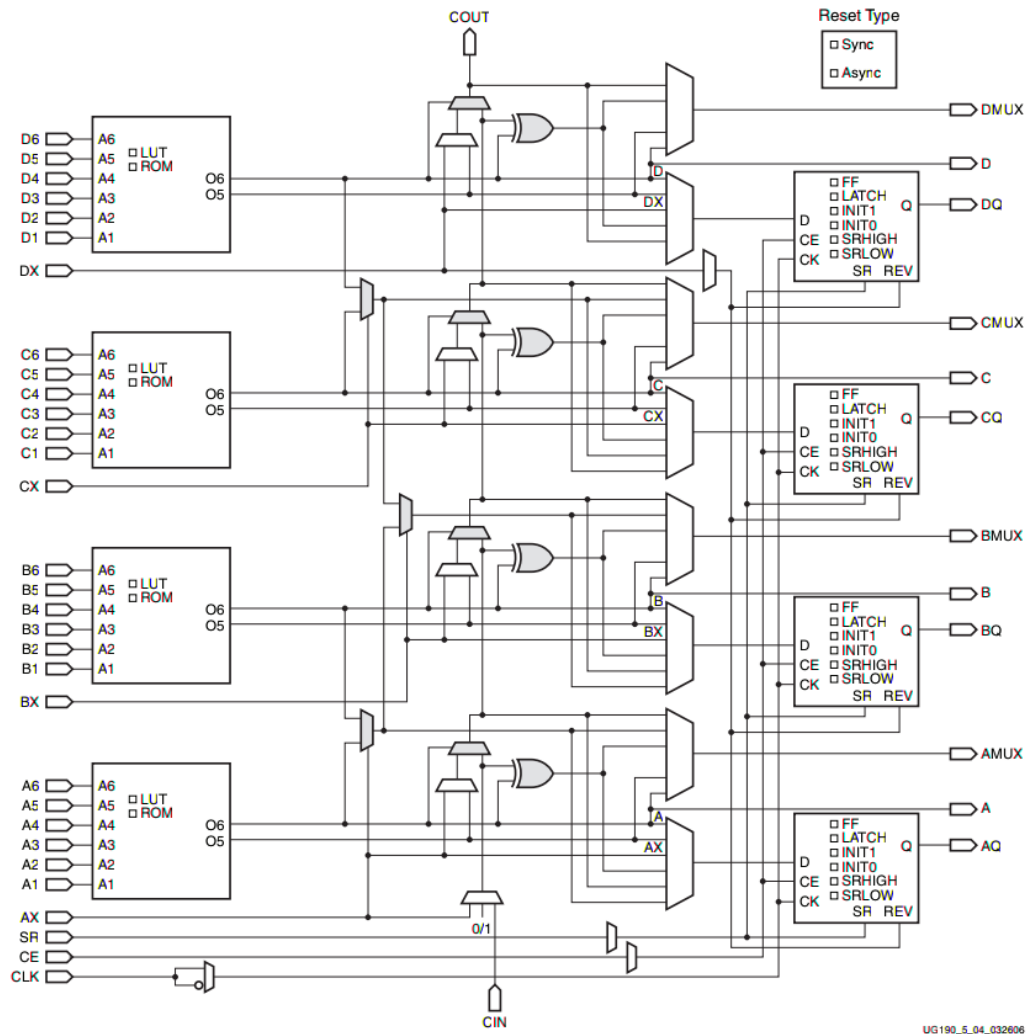


Fig. 1.2 Logic Block (SLICEL) diagram from Virtex-5 device of Xilinx

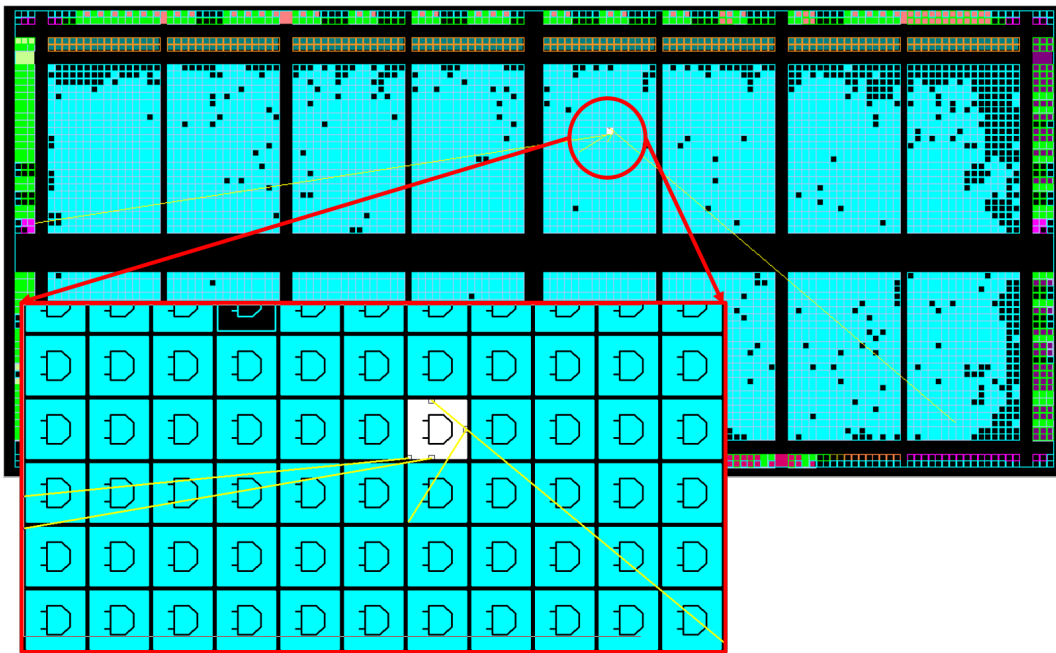


Fig. 1.3 A routed design mapped on ProASIC3 from Microsemi

Place and Route tool is used to map the design to the hardware resources on FPGA. In this process, the netlist could be used for different level simulation to validate the design and timing correctness. Afterwards, the Post-Layout netlist is used by Bitstream Generation tool to generate the bitstream file that can be downloaded to FPGA for implement the design.

### 1.2.2 GPGPU

The Graphics Processing Units (GPUs) are special purpose processors devoted to processing a large amount of data in a parallel fashion. Originally, this technology was developed to accelerate image processing applications targeting multimedia applications. Later on, this technology was adapted in other fields, such as High-Performance Computing (HPC) for scientific applications, increasing the throughput and performance. GPUs employed in these fields can be also known as General Purpose Graphics Processing Units (GPGPUs).

GPGPU devices are able to execute multiple tasks concurrently. Internally, those tasks are divided into multiple groups of processes (also known as Work-Groups,

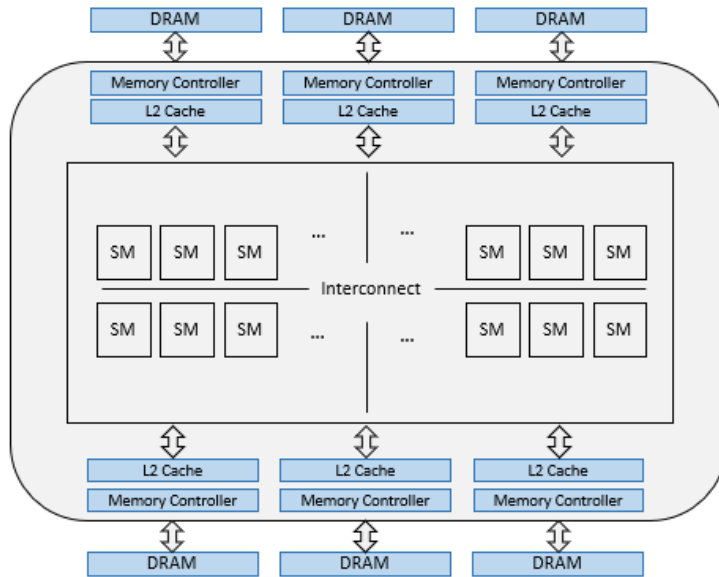


Fig. 1.4 Contemporary GPU architecture

Warps (Thread-Groups or Wavefronts) to be executed concurrently. The size of the groups (32, 48 or 64 processes) depends on the granularity of the device technology.

In general, GPGPUs architecture is based on the Single Instruction-Multiple Data (SIMD) computer taxonomy [?]. This architecture employs multiple identical processing units (or functional units) to perform the same operation on a group of processes targeting different data operands as it is shown in Figure 1.4.

The structure of a modern processing unit (also called Streaming Multiprocessor (SM)) includes an instruction cache memory, associated logic for instruction fetching, decoding, one or more Work-group scheduler controllers and dispatcher units, a register file, multiple integer and floating point units (also known as execution units or CUDA cores), and special purpose accelerators such as modules devoted to matrix calculations and transcendental operations.

Some manufacturers define the processing unit as combinations of two or multiple SMs units. However, the basic operation of the system is the same. This operation is controlled by the SM scheduler controller and it is in charge of starting the execution of the SM by searching a work-group and dispatching to the SM.

The SM execution starts by searching and decoding an instruction from the instruction cache memory. Then, each Work-group element (process or thread)



search the data operands in memory followed by the operation on the execution units (CUDA cores). Finally, results are stored in memory locations and a new instruction is dispatched.

Nowadays, GPGPUs technologies are promising processing solutions in complex and safety-critical applications, such as autonomous and semi-autonomous vehicles. Moreover, those devices are designed and implemented employing aggressive technology scaling approaches in order to fulfill performance and power constraints. Nevertheless, it is well known that those integration technologies are more prone to suffer from external effects such as radiation.

# **Part I**

## **Single Event Transient**



## **Chapter 2**

# **Basic Mechanism of Single Event Transient**

Advanced digital circuits play an important role in a growing number of applications. When used in mission critical application, digital circuits require a special attention to the dependability aspect. In fact, one of the most critical environment aspect that could lead to the failure of modern integrated technologies and systems is radiation. Radiation effects on VLSI technology are provoked when radiation particles deposits a charge. If the transited charge is enough to create a voltage glitch inside the circuit, a spurious voltage glitch will be generated defined as Single Event Transient (SET). Once the voltage glitch is generated, it may propagate through the circuit. If the pulse reach to the storage element of the circuit and be captured by the storage element, it may reach to the primary output of a circuit. Thus, provoking a functional interruption. Moreover, considering the device technology shirking process makes them more prone to be affected by a radiation particle. Likewise, considering the increasing of the clock frequency of the recent complex design, the probability of sampling SET pulse is growing. Therefore, the SET phenomena is becoming more critical for recent technologies [8].

### **2.1 From radiation particle to voltage pulse**

The Single Event Transient (SET) effects in nanoscale devices are usually the consequences of charge particle strike [9]. When a highly charged particle interact with

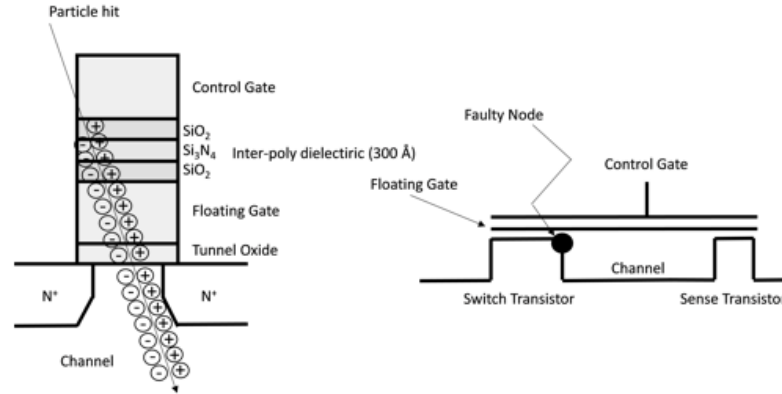


Fig. 2.1 Floating gate transistor layout in the 130 nm Flash-based FPGA and the correspondent sensitive node generating transient pulses

the silicon junction of the device, the produced free mobile carriers are concentrated within the depletion region of a p-n junction in one of the transistor sensitive nodes. Therefore, the SET pulse will appear in the drain of the transistor because the ionization can charge or discharge the active area. Figure 2.1 present a floating gate transistor, however, the SET can happen in a regular transistor too.

As a result of the interaction of particle within the device, a voltage glitch is generated known as Single Event Transient (SET). Moreover, the characteristics of the generated SET pulse depends on several elements. To elaborate more, the SET shape depends on several factors including the incident particle, the device technology node it strikes, LET of the particles, the incident angle and the presented electrical field at the incident moment [6].

### 2.1.1 Basic mechanism of Single Event Transient

Transient errors are happening due to two major concerns: Firstly, if a storage element junction is affected, it may cause a flipping of the original logic value of the storage element, knowing as Single Event Upset (SEU). Under other condition, if the particle hits a transistor as a part of combinational logic, it may generate SET voltage pulse [10]. The generated pulses have the duration between picosecond and nanosecond. The typical pulse shapes generated by heavy ion particles are depicted in Figure 2.2.

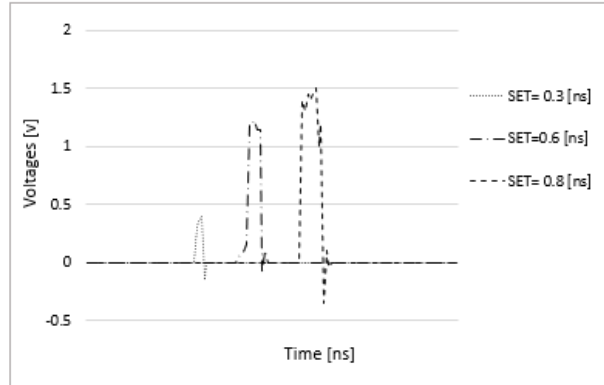


Fig. 2.2 Examples of transient pulses generated by our electrical fault injection platform and mimic heavy ions radiation particles hitting the sensitive nodes of a 130nm Flash-based FPGAs

Under other condition, if the particle hits a transistor as a part of combinational logic generating a SET, the pulse can propagate through routing and logic resources in the circuit until it is captured by a sequential element, typically a Flip-Flop (FF), causing a bit error. This error is propagating its effect during the circuit execution [10].

### 2.1.2 SET life-cycle inside the device

When a sensitive node of a logic cell is hit by a highly charged particle, a voltage glitch is generated inside the logic cell. The induced pulse may propagate through the logic depending on the FPGA tile configuration. The pulse may directly cause a SEU if the tile is configured to implement a latch or Flip-Flop. On the other hand, if the tile is configured to implement a logic gate, in a condition that the voltage amplitude of the induced pulse is more than  $V_{dd}/2$ , the pulse may propagate along the path. During the propagation, depending on the type of the logic which the pulse is propagating through, as an example inverting gates (INV, NAND, NOR,...) or non-inverting logic gates (AND, OR), the features of the SET such as shape, width, voltage amplitude and the propagation velocity may undergo different changes [11] [12]. As a result, the pulse may either be masked due to other gate inputs along the path or eventually reach a storage element, being captured by a storage element. Then, the corrupt stored value leads to misbehavior of the implemented design.

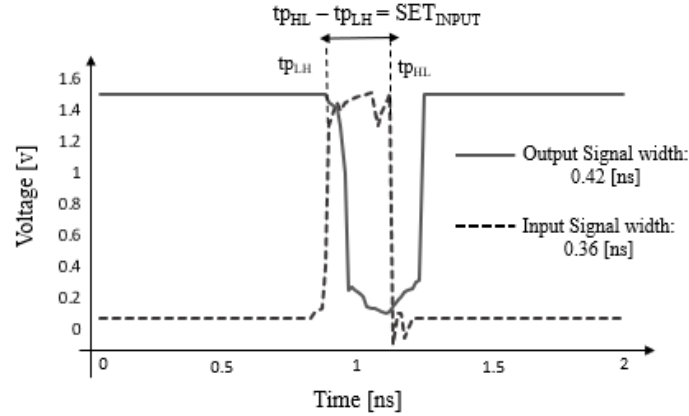


Fig. 2.3 Propagation of SET pulse through an Inverter with an input transition of 0-1-0

Several methods are dedicated to investigate the behavior of SET pulse propagating inside the circuit implemented in the device [13]. The results of these investigations show that SET pulse could be either filtered or broadened while traversing different logic gates, which means that the characteristics of the pulse, such as amplitude and width of the SET pulse, at the input of the storage element are dependent on the number and type of gates along the propagation path. Figure 2.3 represents an example of a SET pulse traversing through an Inverter gate with a logic transition of 0-1-0. In this figure,  $t_{pLH}$  and  $t_{pHL}$  are representing the propagation delay of an Inverter as well as  $\Delta t$  which shows the different between them.

As it can be observed, due to the delay unbalance at different circuit nodes (different between propagation delays of  $t_{pLH}$  and  $t_{pHL}$ ), the transient pulse is facing a broadening or filtering effect, known as Propagation Pulse Broadening (PIPB) effect. To elaborate more, if the delay for propagating the first transition is shorter than the delay of propagation of the second transition, the SET pulse is broadened which means that the duration of the SET pulse is increasing which increases the probability of SET pulse captured by storage element. On the other hand, if the delay for propagating the second transition is shorter than the delay for propagating the first transition, the SET pulse is attenuated which means the duration of the SET pulse increases which reduce the possibility of sampling SET pulse by storage element.

## **Chapter 3**

# **Analysis of Single Event Transient**

The aggressive scaling trend in recent technologies makes Single Event Transient (SET) one of the main critical faults within electronic circuits. The decreasing of device and interconnect dimensions and reduction in the node capacitance of the circuits leads to the generation of SET pulses even with a low energy particles. On the other side, because of the high working frequency of recent circuits, the probability of the generated SET pulses being catch by storage element is increasing. Therefore, SET pulses are becoming more and more critical issues.

Considering recent technologies, two main FPGA families are the main interests specially for space mission applications: Flash-based FPGAs and SRAM-based FPGAs. Since the configuration memory cells of Flash-based FPGA are immune to Single Event Upsets (SEUs) [14], SETs in user configurable resources are the major source of soft-errors. As a result, several studies are dedicated to the characterization of the behavior of SETs in Flash-based FPGAs. On the other side, SET behavior in SRAM-based FPGAs is not studied in deep. Therefore, in order to provide a comprehensive characterization of behavior of SET pulse in the recent FPGAs, we focused on the both technologies, Flash-based FPGAs and SRAM-based FPGAs.

### **3.1 SET Characterization**

In order to provide the comprehensive characterization of SET pulses in modern technology, we focused on both Flash-based and SRAM- based FPGAs. As a first step, for generating SET pulses, electrical injection method has been used in order



to provide the complete control of the generated pulse parameters. The generated pulse has been inserted in several circuits implemented on the FPGA in order to evaluate the behavior of SET pulse propagating through different logic gates and routing interconnections. The results have been provided by analysis the SET pulses reaching to the output of the implemented circuits with respect to the generated source SET pulse.

### 3.1.1 Electrical SET Injection

In order to emulate Single Event Transient Pulses generated by radiation particles, there are three main methodologies. The first one is radiation test. Radiation test is providing the most realistic conditions with respect to the real radiation environment. However, it is an expensive test. Moreover, it is not possible to control the parameters of generated source SET neither the location of the radiation incident. Therefore, it is providing the random incident which is close to reality but not sufficient enough for analyzing the features of the SET pulse accurately. The second method is laser test. Laser test is less expensive comparing to radiation test but still knows as an expensive test methodology. Moreover, some specifications of the technology under the study is required which is not always available. One sufficient and profitable method to generate a voltage pulse is electrical injection [13]. As the benefits of using this method, the possibility to control the parameters of the generated pulse and also the location and time of injection of the pulse can be mentioned .

Electrical injection can be performed in two methods: External electrical injection and internal electrical injection. The adoption of external electrical injection is running down, since the distortion effects would affect the generate pulse traversing the input ports. We choose internal electrical injection to generate SET pulses internally. Therefore, we avoid the filtering effect of IOs structure which leads to a better control of the pulse parameters. The logical scheme of internal pulse injection generation is represented in Figure 3.1. Please notice that the proposed logic scheme of internal injection is adoptable to both FPGAs and ASICs. However, ASIC technology is out of the scope of the this dissertation.

The logical scheme includes "Inverter" and "AND" gates. The input of the Inverter chain is connected to Signal Generator while the last cell of the Inverter chain is connected to one of the "AND" input. The other input of the "AND" gate is

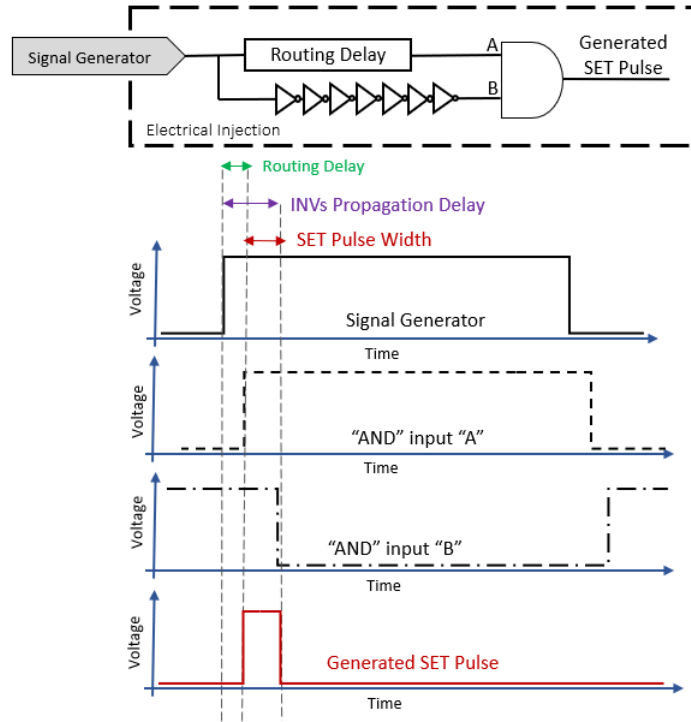


Fig. 3.1 The developed logic scheme of internal injection generation

connected to the Signal Generator directly while the output of "AND" is connected to the connected injection point internally. Due to difference of the delay between the signals reaching to Inputs A and B of "AND" gate, the SET pulse will be generated at the output of the "AND" Figure 3.1. The duration of the generated pulse as an output of the "AND" gate is proportional to the delay of the number of Inverters used in the chain. The longer is the Inverter chain, the wider is the SET pulse, represented by Equation 3.1.

$$\Delta_{SET} = \Delta_{INV} - \Delta_{Route} \quad (3.1)$$

### 3.1.2 SET propagation characterization

Several studies have been dedicated to analyze the characterization of radiation-induced SETs in advanced digital circuits [15]. Most of them are focused at the simulation level including technology-based physical equation for the evaluation of the radiation incident and the propagation through the circuit [16]. A new methodology for measuring the duration of the radiation-induced SET pulse is introduced in [11] and [17]. However, the propagation of SET pulse has been studies

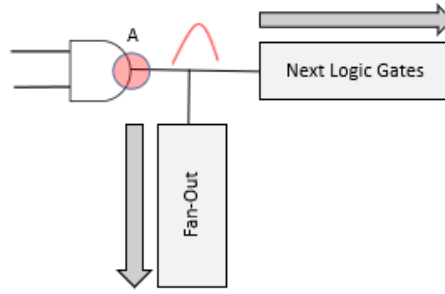


Fig. 3.2 A Logic sensitive node and the two observability methods: towards next logic gates and toward fan-out

only considering the delay of SET pulse propagating through the circuit and the filtering and broadening of the pulse known as PIPB effect is not considered. Some research works report the radiation test experiment of SET propagation on custom circuits design for triggering and monitoring SET pulses [18]. These works show a strong SET pulse width modulation when SET pulse is propagating through logic gates and routing interconnections. Moreover, it has been observed that SET pulse width at the input of the storage element is strongly dependent to the technology under the study [11]. Therefore, it becomes mandatory to study and analyze the behavior of SET pulse in different technology.

In this section, we move toward details of SET life-cycle inside the circuits and propose an approach to evaluate it. The fundamental of this approach is the fact that any kind of SET generated in sensitive nodes and propagated in different position encounter different points. At any point of its logic gate traversing, the SET propagation is dependent on two main factors: The logic gates in front of the pulse starting from the sensitive point under the study and the fan out gates presents in a give position during the pulse propagation, represented in Figure 3.2.

Considering that node A is the typical sensitive node of the implemented design under the study, the SET pulse at this location experiences two different phenomena: The first one is regarding the SET originated in a different position, propagated thought the circuit and reaching to this node. The second is when the particle incident occurs at this location and creates a SET pulse. Independently of the pulse originated at this location or propagated up to here, the future life-cycle of SET pulse at node A is dependent to two features: The logic gates in front of a the SET pulse and the fan-out of node A. To elaborate more, SET pulse at node A faces a chain of

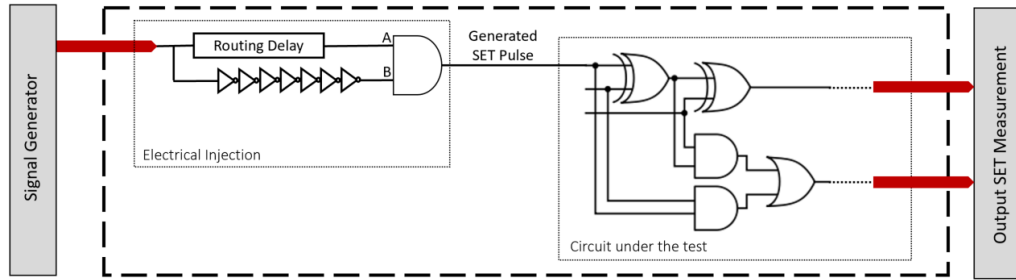


Fig. 3.3 Scheme of internal electrical pulse generator

gates in front with different propagation behavior which is dependent to the type and physical position of the logical gates of the used technology. Moreover, it observes the fan-out connected to the node.

In order to evaluate the life-cycle of SET from its generation to the reach of a destination point such as Flip-Flop or an I/O pin, we design a test setup. In this setup, the SET pulse has been generated using internal electrical injection. The generated SET has been propagated through the designed circuit. At the end, the duration of the propagated pulse has been measured. Figure 3.3 represent the scheme of this evaluation setup. This evaluation setup has been applied to two different technologies: Flash-based and SRAM-based FPGA.

### 3.1.3 SET characterization test setup on Flash-based FPGAs

Flash-based FPGAs are known as the golden core for aerospace application. However, when used in mission critical application, reliability of these devices requires a special attention. Considering aerospace application, radiation particles such as neutrons, protons or heavy ions can hit the sensitive region of the device which may be within the layout layers of a gate or within routing segment of electrical buffers. As a result, a voltage glitch will be generated which may reach the primary output of a circuit causing a functional interruption, depending on different aspects:

1. Logical masking: the propagation of SET pulse may be halted due to the logical behavior of a gate. Therefore, the SET pulse life terminates without affecting the functionality of the gate or circuit.

2. temporal masking: Presenting a scenario while the SET pulse is propagating and reaching to the input of a Flip-flop but not in-time considering the sequential element latching window. Therefore, the SET life will be terminated without being sampled by Flip-flops.
3. Electrical broadening: the SET pulse may go through some modifications where the width and amplitude of the pulse is reduced and make it negligible.

Single Event Transient on Flash-based FPGA technology has been widely investigated with various methods starting from fault injection by simulation to radiation testing. Several research activities focused on the characterization of radiation-induced SETs in sequential and combinational circuits [19]. Several studies have been done at the simulation level, including technology-based physical equation for the evaluation of the radiation strike and its propagation across the technological cell. In [16], the propagation of the transient pulse through the combinational logic data path and routing resources of Flash-based FPGA has been evaluated. New insight on Flash-based FPGA is investigated in [20]. A new method for measuring the width of radiation-induced transient faults is introduced in [11] and [17]. However, they are not representing a realistic design. Because these methods evaluating the effect due to the delay of SET pulse without considering the filtering and broadening effects. Recent studies reported radiation test experiment and electrical fault injection of SET propagation on custom circuits designed specifically to observe SET pulses [18].

### **Flash-based FPGA test setup**

In order to evaluate the life-cycle of SET pulse, from the generation to the storage element, we monitor the behavior of SET on Flash-based technology by configuring different test setups while the SET pulse has been generated using internal electrical injection.

Four types of test setups have been considered for the SET characterization with respect to typical circuit design. The first scenario represents the combinational logic, considering a chain of logic gates. For the second scenario, a fan-out has been added to the logic chain. The third and fourth scenarios are representing a more detailed cases where the existence of divergence and convergence of combinational node are considered.

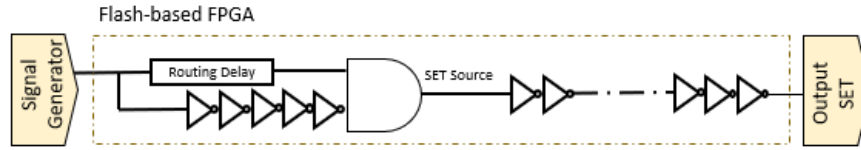


Fig. 3.4 Logical scheme of the test- scenario 1

The test setups have been implemented on Microsemi ProASIC3 A3P250 Flash-based FPGA. The results have been classified in terms of ratio between the output SET pulse at the end of the chain while the source SET generated by internal injector has been inserted at the start of the chain. In order to generate the SET pulse, a pulse generator has been used which includes features such as : 330 MHz, selectable pulse pattern modes and logic test clips. For measuring the pulse at the output of the FPGA, an Oscilloscope equipped with high-impedance calibrated probes with features of: Bandwidth of 100 MHz, 2 Channel, Sample rate on each channel equal to 1.0 GS/s, input sensitivity range from 2mV to 5V/div and USB interface. Please notice Oscilloscope has been used for measuring the SET pulse generated at the output of the Internal electrical injection. For all the experiments, the source SET pulse is considered equal to 3.5 ns.

### First scenario: analysis of Inverter-string

The first scenario is dedicate to the characterization of the logical gates using internal electrical injection for generated SET pulse. Therefore, we used Inverter gates as logical gates while we used different length of Inverter-string. The logic scheme of this scenario is presented in 3.4.

The SET has been generated using signal generator connected to internal electrical injector. The generated SET pulse is connected to the first gate of the inverter string internally while the SET reaching to the last gate in the chain has been monitored.

The internal electrical injector had been implemented considering the fix placement and routing characteristics. The same as the placement of the logical gates which has been performed following the same distances between the logical gates in order to guarantee the similar distance between the logic cells. This routing and placement rules is shown is Figure 3.5.

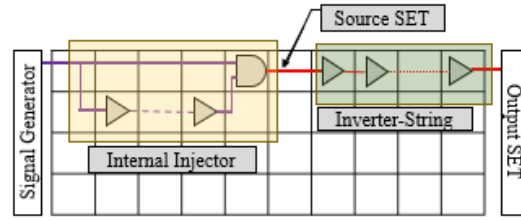


Fig. 3.5 An overview of the placement layout- Scenario 1

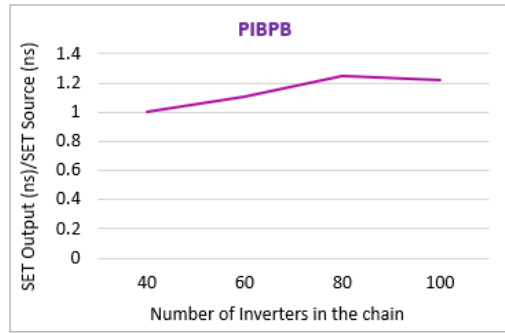


Fig. 3.6 Propagation Induced Pulse Broadening- Scenario 1 (Source SET = 3.5 ns)

As a first test setup, four designs with a set of 40, 60, 80 and 100 inverters in the chain have been considered. This logic chain has been manually placed side by side in the array with minimal distance connection between each VersaTile stage. Please notice that the placement for internal injector components has been fixed for all the experiment. We inject SET in the first Inverter of the chain and we measure the SET at the output of the chain. The results have been classified in terms of ratio between the width of output SET and the source generated SET pulse. This results has been shown in Figure 3.6. As it can be seen, by increasing the number of Inverters in the chain, the PIPB value is increasing until it reaches a saturation point. Moreover, we also measure the delay between the source SET and the output SET pulse. As it can be observed in Figure 3.7, the delay is increasing linearly with respect to the number of Inverters used in the chain.

### Second scenario: analysis of Inverter-string and fan-out of Inverters

The goal of the second scenario is to evaluate the effect of fan-out on SET propagation behavior. Therefore, the second scenario is an extension of the first one considering the inclusion of various logic gates as fan-out. The fan-out has been connected to the

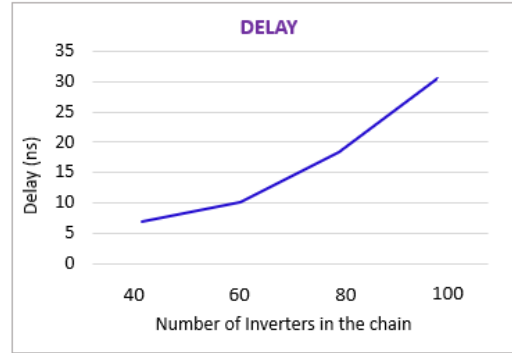


Fig. 3.7 Inverter chain delay- Scenario 1

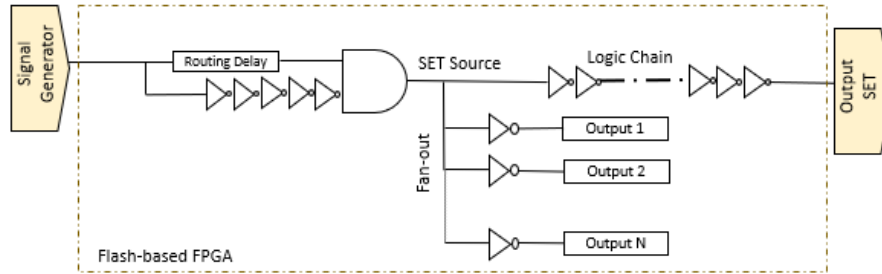


Fig. 3.8 Logical scheme of the test- Scenario 2

beginning of the chain where the output of the internal injector is connected while the SET in the output of the chain has been monitored. Figure 3.8 represents the logic scheme of the second scenario.

The same as Scenario 1, the design of place and route of the fan-out follows the minimal distance routing between each VersaTile. This placement design is fixed for all the performed experiment for the second scenarios which has been shown in Figure 3.9

Please notice that the output of each fan-out is tied to the output pin. Therefore, it will not be simplified with the design tool and this output pins has been fixed for all the tests.

In order to implement the second scenario, we extended the first setup by adding different number of 20, 40 and 60 Inverter gates as fan-out connected to the different number of 40, 60, 80 and 100 Inverters. For each combination of Inverter string and fan-out, the result related to the PIPB effect has been denoted in Figure 3.10. As it can be observed, by increasing the length of the Inverter string, the PIPB increased



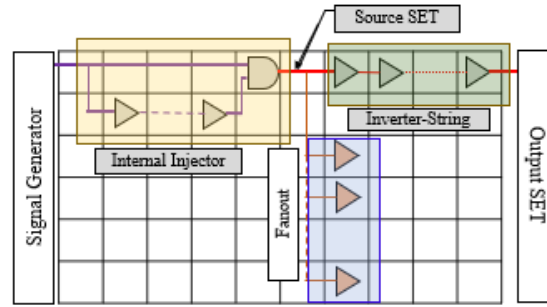


Fig. 3.9 An overview of the placement layout- Scenario 2

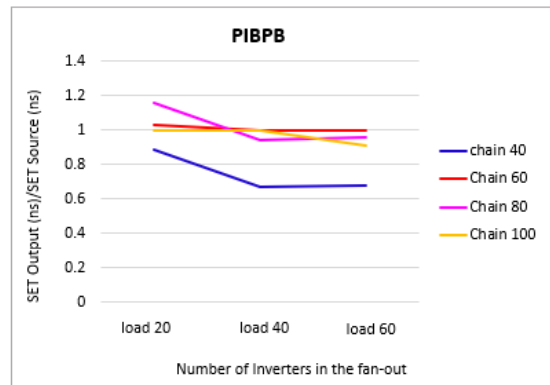


Fig. 3.10 Propagation Induced Pulse Broadening(PIPB)- Scenario 2 (Source SET = 3.5 ns)

which verifies the result of the scenario 1. More importantly, it can be observed that the PIPB is progressively attenuated by increasing the fan-out for a fixed number of Inverters as a chain.

More interestingly, considering the delay of the circuit, we measured the delay between the delay of the input SET and the output SET. It has been observed that for fixed number of Inverters in the chain, by increasing the number of Inverters in the load, the delay of the circuit is not increasing and it is fixed, represented in Figure 3.11.

Therefore, considering fan-out as the filtering architecture, it is possible to modify and decrease the PIPB value without introducing a delay for the circuit.

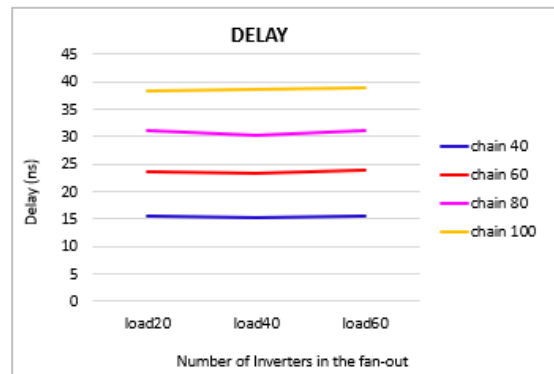


Fig. 3.11 Inverter chain delay- Scenario 2

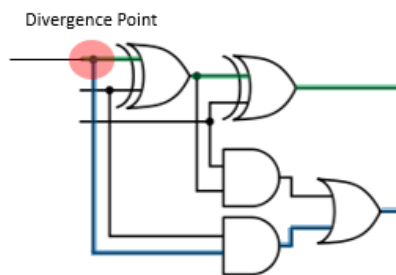


Fig. 3.12 Logical scheme of full adder by SET in the divergence point

### Third scenario: analysis of chain divergence

As a next scenario, we considered the conditions of typical logical designs and behavior of SET while there is an occurrence of divergence of combinational paths. Considering the typical full adder structure represented in Figure 3.12, when there is an occurrence of SET pulse in the divergence point, independent from the fact that SET pulse has been generated in the divergence point or propagated up to this point, the SET pulse spreads through divergence point and propagate through secondary logic string and reach to the storage elements connected to the affected string.

In order to evaluate this condition, the second scenario has been modified to mimic the SET characterization in the divergence node. In the new test setup, we create a divergence node in the middle of the main string and adding the second string to this divergence node. The internal injector has been connected to the input of the main Inverter string while the SET at the output of both paths have been monitored. The same as previous test setups, the placement and routing of the implemented

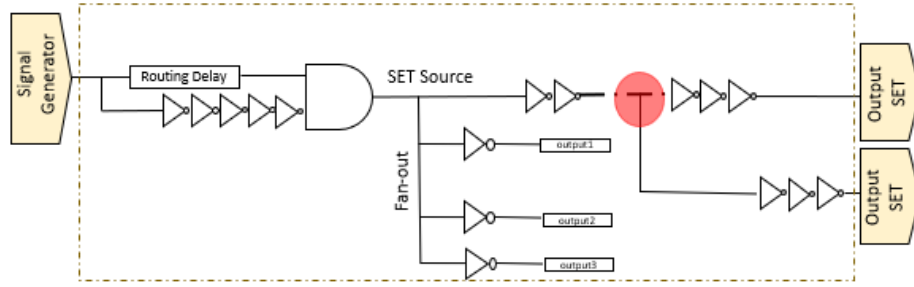


Fig. 3.13 Conceptual scheme- third scenario

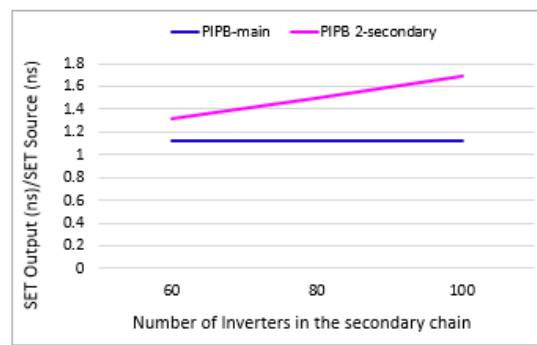


Fig. 3.14 PIPB report- third scenario (Source SET = 3.5 ns)

design has been controlled totally and tied to the same placement for all the tests. The scheme of the third scenario has been presented in Figure 3.13

In order to study the behavior of SET in the divergence point of the implemented design, the length of the main string has been fixed as 60 Inverters and different number of 60, 80 and 100 Inverters have been tied to the secondary string and the fan-out has been considered as 60 Inverters. The propagated SET pulses reaching to the end of the main string (First output) and the secondary string (Second output) have been monitored and measured. Figure 3.14 reports the PIPB value while Figure 3.15 reports the delay value.

#### Fourth scenario: analysis of chain convergence

The fourth scenario is dedicated to the condition when the SET traverses through divergence point, propagate through different logical paths and routing, reaching to the convergence point of the circuit. Figure 3.16 represents this condition in full adder logical scheme.

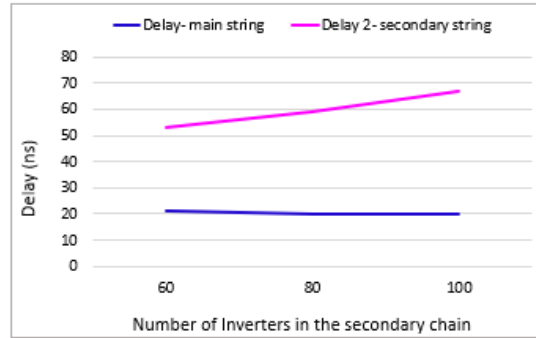


Fig. 3.15 Delay report- third scenario

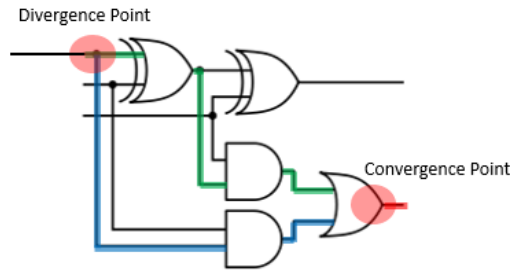


Fig. 3.16 Logical scheme of full adder affected by SET in the convergence point

In order to mimic this phenomenon, we modify the previous scenario, adding a convergence node to the test design. Figure 3.17 shows the logical scheme of the design. The input of the design is connected to the Signal Generator while the generated SET goes through divergence node, multiplying in two and propagating through two paths of A and B. The propagated SETs reach to the convergence point and merge together. The SET reaching to the convergence point is monitored in order to characterize its propagation exclusively. Please take in to consideration that the placement and routing of the electrical injection, string and fan-out have been fixed.

In order to analyze the behavior of the SETs while it has been multiplied into two SETs through divergence point of the design, propagated and reached to the convergence point, we consider a string of 20, 40 and 60 Inverters and fan-out of 20, 40 and 60 Inverter gate. The SET has been injected at the start of the string and SET propagated through two designed paths. The first one is a string of Inverters while the second one consists of only routing segment. The SET pulse reaching to the convergence point has been observed.

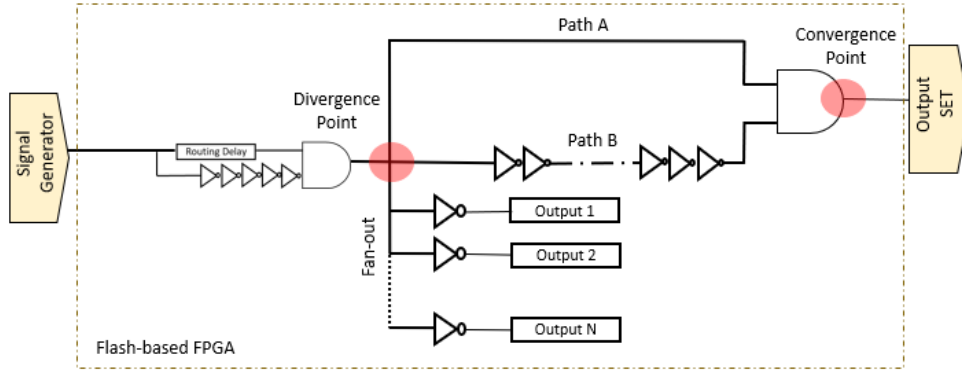


Fig. 3.17 Conceptual scheme- fourth scenario

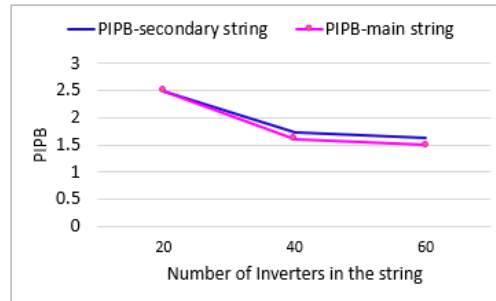


Fig. 3.18 PIPB report- fourth scenario (Source SET = 3.5 ns)

Figure 3.18 presents the obtained results regarding obtained PIPB while Figure 3.19 reports the delay of the circuit. The results shows that the respective counterpart only marginally influences the PIPBs of two strings.

Moreover, due to merging and overlapping of the two propagated pulses at the Convergence point, a new phenomenon is observed. Due to this phenomenon, two results can be observed. The first one happens when there is a large difference on the propagation delay of the two separate pulses, represented in Figure 3.20(a). The second happens when the difference between the delay of the two paths is small, a converged SET can be observed which is characterized by an extremely large width, Figure 3.20(b). This phenomenon is explained in details in the following sections.

### 3.1.4 SET characterization test setup on SRAM-based FPGAs

When applying FPGA to mission critical application used in space environment, two main families are considered: Flash-based FPGA and SRAM-based FPGA. Due

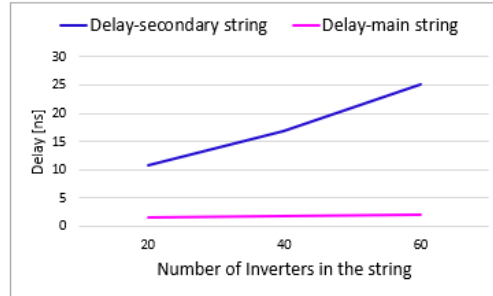


Fig. 3.19 Delay report- fourth scenario

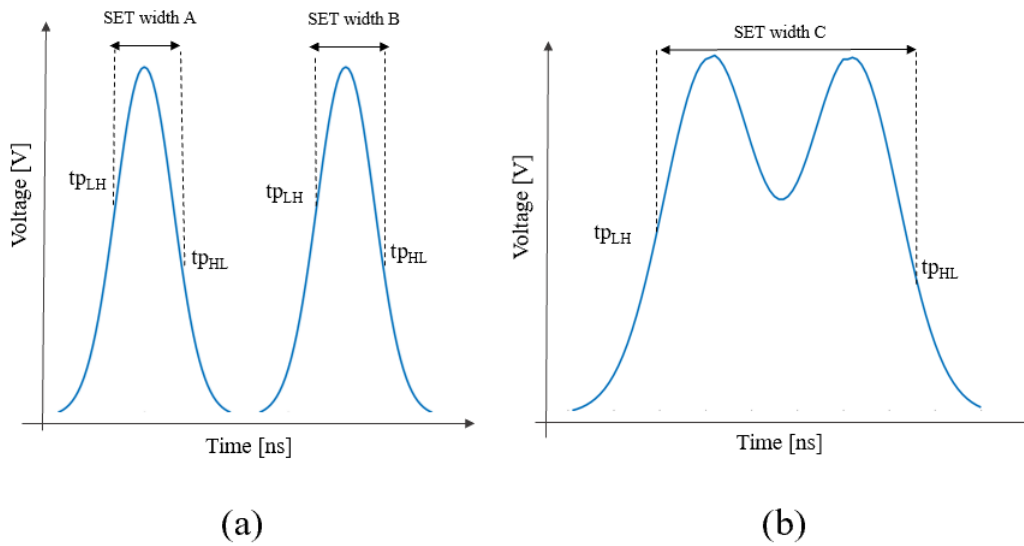


Fig. 3.20 An example of SET propagated through two convergence paths and generating: two independent SET pulse (a) and C-SET pulse (b)

to the fact that configuration memory cells of Flash-based FPGAs are essentially immune to Single Event Upsets (SEUs) [21], SETs in configurable resources are the major concern of soft errors. Therefore, life-cycle of SET in Flash-based FPGA has been studied several times while the SET pulse in SRAM-based FPGAs has not been investigated widely. In [14], a new methodology for generation and measurement of SET has been proposed. SET pulses propagating through different logic chain have been studied while the method is not tested in realistic circuits. Moreover, this methodology did not cover a comprehensive characterization of SET through the circuit since it was not applying an internal injection method able to generate a transient pulse. Therefore, we used internal electrical injection capable of generating SET pulses at various amplitude and width within the SRAM-based FPGA. Moreover, the measuring methodology has been applied which allows measuring of the propagation pulse shape with respect to different input pulse width for any type of logic function implemented on SRAM-based FPGA LUT. The methodology has been applied to circuits implemented on SRAM-based FPGA in order to confirm the efficiency of the developed work flow for providing the comprehensive characterization of SET on SRAM-based FPGA.

The same as Flash-based FPGA, internal electrical injection has been used to generate SET pulses internally in SRAM-based FPGA, Figure 3.1. Therefore, controlling the width of generated SET pulse is acquired by changing the number of Inverters or constraints the placement of the gates involved in the internal SET generator.

### **Characterization analysis test**

In order to evaluate the life-cycle of SET pulse, starting from the injection point to the storage element or output of the circuit, different test setups have been designed. For these test setups, different logic gates have been considered while the SETs with different pulse duration have been generated. The test setups have been designed with respect to typical circuits. Therefore, different combinational logics such as INV, AND, NAND, OR, XOR and XNOR have been considered for generating chain of logic gates. The output of internal electrical injection has been connected to the beginning of the chains configured with different logic gates. The SET pulse is propagating through the chain and at the end of the chain, the propagated SET pulse

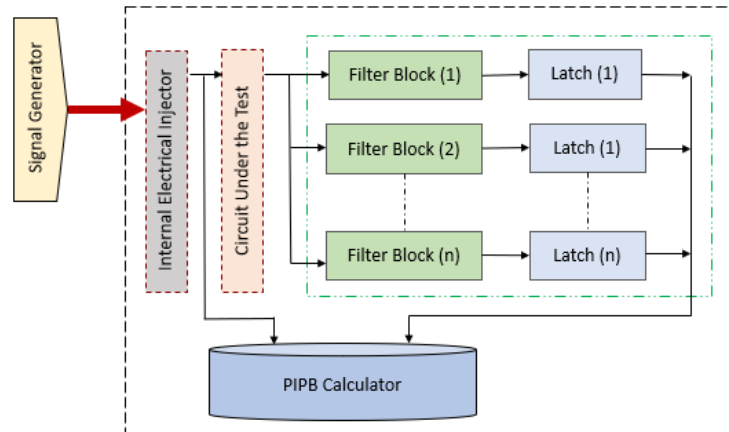


Fig. 3.21 Overview of the global analysis methodology for SRAM-based FPGAs

is monitored. The logic scheme of the generated test setup is observed in Figure 3.21.

An automatic methodology for measuring output SET and PIPB coefficient has been developed, which process the generated pulse, measures the propagated pulse width and computes the PIPB coefficient for different gates. This procedure is repeated until the error rate is lower than a threshold which is defined by users. For calculating the PIPB coefficient, Monte Carlo approach is applied while the number of iteration has been sets to maximum of 10K iterations. As a reference error, we used the standard signal-error lower than  $10^{-5}$ .

Please note that the placement and routing of the chains composed of different types of gates in each configuration has been controlled strictly in order to guarantee the same distance between each cells and this the same delay on the routing paths in order to calculate the PIPB coefficient value for each individual gate of different types with the information regarding the injected SET pulse and the captured SET pulse at the end of the chain.

### Capture and monitor of SET Pulse

In order to measure the width of the SET pulse reaching to the end of the chain, two methodology can be applied. The first one is using measurement equipment such as oscilloscope for measurement of the width of the pulse, as it has been used for Flash-based FPGA setup. The second method is to apply the the internal measurement



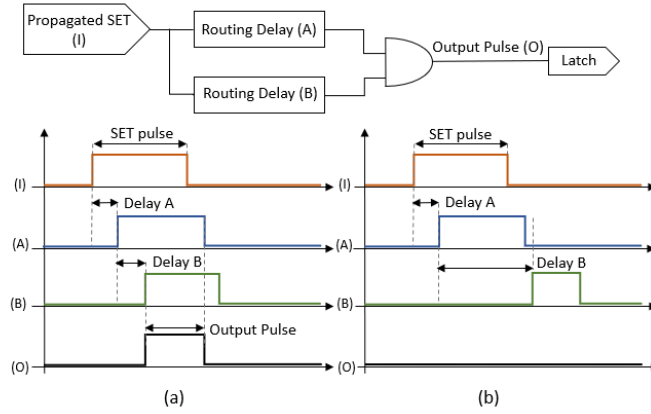


Fig. 3.22 Scheme of the SET pulse measurement circuit

setup such as internal filter based measurement methodology which provides an accurate measurement of pulse duration.

The filter based measurement circuit is composed of an array of SET filtering blocks where each block is in charge of filtering SET pulses within certain range. Figure 3.22 represents the logical scheme of the filtering block.

This methodology works based on the difference between routing delay of path A and routing delay of path B. The SET pulse that has been generated with internal electrical measurement is propagated through the developed chain of logic gates and reaching to the input of the measuring block. The SET pulse at the input of the measuring block will propagate through both path A and B. If the difference of routing delay between path A and B is less than the duration of SET pulse, the pulse propagates through the AND gate and sampled by the latch, presented in Figure 3.22, a. On the other hand, if the difference of the routing delay is more than the width of the pulse, the pulse is filtered and the latch output remain the same, as shown in Figure 3.22, b.

By tuning the difference of the delay for each filtering block in an array of such similar blocks, the width of SET pulse can be monitored through decoding the output values of all the latches.

### Experimental setup on SRAM-based FPGA

Xilinx KC705 development board adopting a Xilinx-7 SRAM-based FPGAs has been used to evaluate the SET behavior of 28nm technology. As a primary step, a simulation environment has been developed. The simulation environment is used for simulating different chains of gates, while injecting SET pulses at the input and observing the output SET pulses. For the simulation environment, the focus is on the timing analysis in order to confirm the necessity of using SET injection for performing characterization on SRAM-based FPGA.

As a second step, Xilinx Vivado Design implementation tool is used to develop chains of different gates of INV, AND, NAND, OR, XOR and XNOR. Please note that the concept of *Gate* is not introduced in SRAM-based FPGA. Instead, Look Up Tables(LUT) are considered to perform the required functions. We inject SET pulses with different widths at the input of the chain using internal electrical injection and measured SET pulses at the end of the chain. The results has been reported in terms of ratio between the SET measured at the end of the chain and the injected SET at the start of the chain knows as PIPB coefficient.

### Simulation results

A simulation environment is used to assess the necessity of using SET pulse injection and measurement for evaluating PIPB effect on SRAM-based FPGAs. The result confirmed that SET pulse reaching to the end of the chain is facing a propagation delay which is dependent to the length of the chain. Moreover, the width of the SET pulse is equal to the injected one at the start of the chain and it is not affected by broadening or filtering effect. Figure 3.23 represents the simulation result related to a chain of 100 INVs while the SET pulse of 0.2 ns has been injected at the start of the chain and the pulse with the same width has been observed at the end of the chain with the propagation delay. Therefore, it can be concluded that the commercial simulation tools are not accurate for providing information regarding the behavior of SET propagation in chain of gates. Therefore, developing a work flow is for evaluating the behavior of SET pulse is necessary.

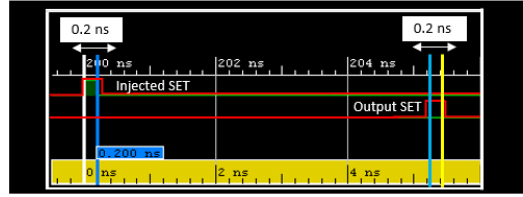


Fig. 3.23 Simulation results for injecting SET of 0.2 ns to a chain of 100 INVs

Table 3.1 Generated SET using internal electrical injection.

Pulse Generator	Pulse Width[ns]
Pulse 1	0.4
Pulse 2	0.8
Pulse 3	1.0
Pulse 4	1.2
Pulse 5	1.4
Pulse 6	1.8
Pulse 7	2.0
Pulse 8	2.2

### Test setup

Internal electrical injection is used to generate SET pulse. We generate eight different SETs by changing the placement and routing of the SET injector itself. Therefore, we introduce different path delays while the number of used inverters are fixed as three. Table 3.1 reports the widths of the generated SET pulses.

Different test setup including Chain of 100 INV, AND, NAND, OR, XOR and XNOR logic gates have been developed. The generated SET has been connected to the beginning of the chain while the measurement block is connected to the end of the chain. Moreover, no other gate is connected to the chain to avoid any possible distortion during measurement and analysis.

In order to overcome the logical masking of the generated pulse, for the logics with more than one input, the other inputs are all set to unmasked condition such as AND and OR gates. For example, the second input of AND gates is connected to 1, OR gate is set to 0 and NAND gate is set to 0 while for XOR and XNOR, both condition of setting the second input to 0 and 1 have been implemented.

Considering the described SET injector and the measurement block, the widths of SET pulses at the end of each chain have been monitored. The measurement block

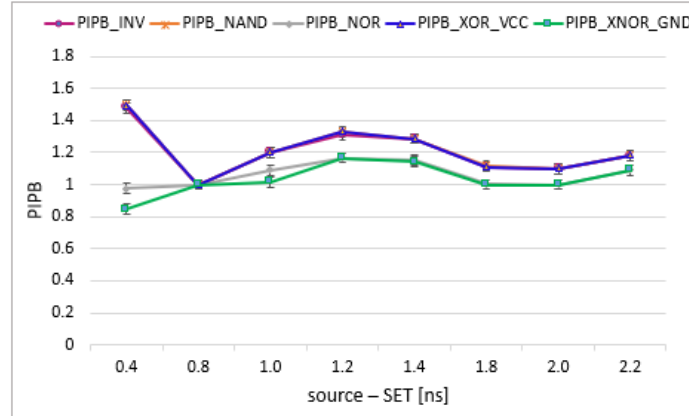


Fig. 3.24 Propagation pulse broadening for inverting gates- each chain includes 100 gates

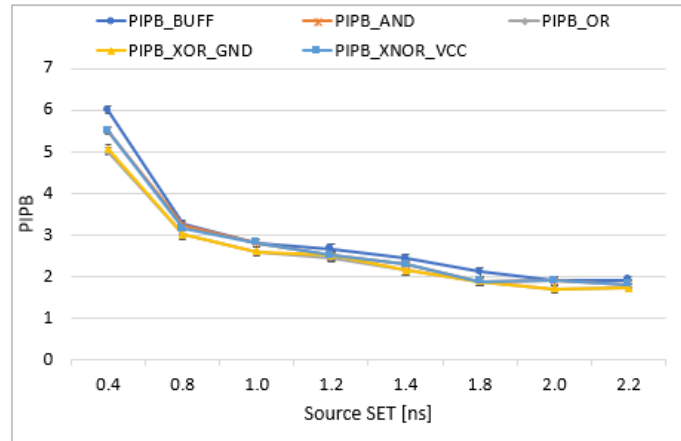


Fig. 3.25 Propagation pulse broadening for non-inverting gates

composed of 32 filtering blocks which is capable of measuring SET pulses with the duration from 200 ps to 6.4 ns with a resolution of 200 ps.

Figure 3.24 represents the results regarding PIPB coefficient for the inverting gates (gates that produce opposite logic value). As it can be observed in the figure, by increasing the width of SET, the PIPB coefficient is progressively attenuated and measured PIPB coefficients for all tested inverting gates are less than 2.

Regarding Non-inverting gates, different behavior is observed. In fact, the results shows that Non-inverting gates show higher sensitivity comparing to inverting gates. Considering Figure 3.25, the PIPB coefficient can reach to 6 for narrow SET pulses, much higher than PIPB of inverting gates. However, when the width of SET pulse increases, the PIPB coefficient decreased drastically.

Table 3.2 Benchmark area utilization.

Circuit	FF[#]	LUT[#]	MUXFX[#]	CARRY[#]	IO[#]
B11	30	108	2	10	15
B12	119	279	8	0	13
B13	51	55	0	0	22
B14	218	2350	50	268	88

Table 3.3 Benchmark critical path resources.

Circuit	Inverting gate[#]	Non-Inverting gate[#]
B11	4	16
B12	8	4
B13	8	2
B14	32	28

### Real test case

Four circuits from ITC'99 benchmark collection have been chosen [22]. The are utilization are reported in Table 3.2.

These four circuits have been implemented on Xilinx KC705 board with the working frequency of 40MHz. For avoiding the erroneous measurement happening due to insertion of the pulse injection infrastructure, the timing and placement constrains has been fixed. Therefore, the implementation tool does not modify the delay characteristics of the logic paths.

For each benchmark, the most critical path regarding timing is selected. Table 3.3 reports the characteristics of the selected path. The eight generated SETs are injected to the first LUT of the critical path while the measuring circuits have been connected to the last LUT of the path.

As it has been shown in Figure 3.26, in circuit B11, by increasing the generated source SET, the PIPB value reduces drastically because the number of non-inverting logics is more than inverting logics. On the other side, for circuit B13, since there are few inverting gates, the PIPB value is not going through the variability due to the PIPB coefficient saturation cause by inverting gates. However, the longer is the path, the higher is the PIPB coefficient as it is expected. It means that the SET pulse will be broadened more when propagating along the path. The measurement error of the PIPB value has been estimated less than 5%.

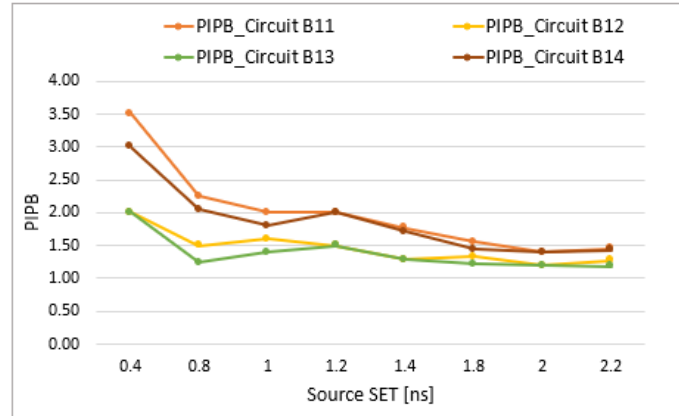


Fig. 3.26 Propagation pulse broadening for circuit benchmarks implemented on SRAM-based FPGAs

As a summary, SET characterization on SRAM-based FPGA is evaluated. Internal electrical injection is used for generating SET pulses while propagated SET pulse is measured using an array of filtering blocks.

### 3.1.5 Research advancements on SET characterization

In order to provide an accurate analysis of SET behavior propagating through the design implemented in different technologies, a behavior of SET pulse in the technology under the study is required. Therefore, a comprehensive characterization of SET pulses is mandatory. Internal electrical injection is used to generate a SET pulse internally and several tests setups have been developed for evaluating the behavior of SET pulse propagating. The test setups are implemented on two main categories of FPGAs: ProASIC3 Flash-based FPGAs and Xilinx KC705 as SRAM-based FPGAs. As a result, a comprehensive behavior of SET propagating through the design is provided including the evaluation of PIPB effect which is a mandatory step for performing an accurate SET analysis.

## 3.2 On the prediction of SETs

In order to apply an efficient mitigation solution to the circuits under exposure, as a first step, it is mandatory to predict the radiation-induced SET phenomena within

the silicon structure of FPGA devices. Some studies have been dedicated to this phenomenon.

In [23], an analytical method for the modeling of SETs has been provided. However, this method is mainly based on the probabilistic calculation of the transient pulse effect which does not represent the correlation of the generated SET pulse with the Linear Energy Transfer (LET) of the radiation particles. Therefore, we propose an effective methodology for prediction of SET phenomena and correlating the type and LET of radiation particles with the generated SET pulses [24]. Thanks to the experimental radiation test that we performed with Heavy-ion particles [25], we developed an environment for evaluating and predicting the generated SET pulses.

### 3.2.1 SET physical dynamic simulation model

The developed platform is based on two analytical models. As a first step, the approach is starting with SET generation models for generating the source SET voltage pulse considering the linear energy transfer of the particle generating SET pulse. The second step is dedicated to physical dynamic model. This model is applied to define the propagation behavior of the voltage pulse applied at the input of the gate, dynamically.

#### SET generation

The incident particle has a scattering path during its penetration in the material, presented in Figure 3.27.

If the energy of the particle is high enough, it can traverse through the material and generate a spurious voltage glitch inside the device. During the interaction of the particle within the device, the radiation particle is transmitting its energy to the device. The dynamic energy of a particular particle incident on a material is defined as its energy loss per unit path length of penetration into the material itself. This correlation has been shown in equation 3.2.

$$Dyn_{LET} = -dE/ds \quad (3.2)$$

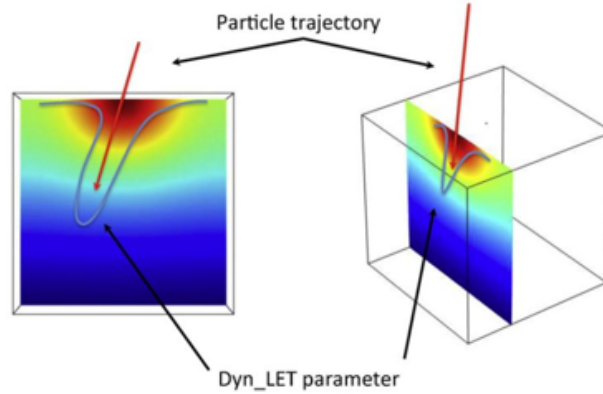


Fig. 3.27 An example of dynamic LET form on a generic empty layout in 2D and 3D .

$Dyn_{LET}$  is defined as the particle incremental energy (dE) loss per incremental distance (ds) traversed in the material. The SET generation mathematical model is correlating the energy of the particle that has been transmitted to the silicon and the maximal depth where the energy has been reached. The overall correlation between the LET and the SET voltage pulse is represented by equation 3.3.

$$\begin{aligned} \Delta V_{pulse}(GATE) &= k_{gate} \times (t_{pLH} - t_{pHL}) \\ t_{pHL} &= \frac{t_{SET}}{Dyn_{LET}} \end{aligned} \quad (3.3)$$

$t_{pHL}$  and  $t_{pLH}$  are the propagation behavior of the related gates while  $\Delta v_{pulse}$  is representing the maximum amplitude of the generated voltage pulse by radiation particles. It is necessary to mention that the maximum voltage pulse is dependent to the transmitted LET and the material which it deposits the energy. This effect has been represented by  $k_{gate}$  coefficient. The dynamic behavior of the developed model is expressed by the ratio between the static time of the SET pulse ( $t_{SET}$ ) divided by the dynamic LET shape and it is calculated by means of a modeled dissipation function created according to each gate layout. An example of dynamic LET calculation has been represented in Figure 3.27, which has been computed base on the radiation particle trajectory, position of the radiation particle and layout topology of the cell. Therefore, the shape of SET pulse analytically depends on the cell sensitive points and from the energy and position of the radiation particle.



### SET propagation behavior

After Generation of SET pulse with respect to LET, the propagation of generated SET pulse has been investigated. Based on the physical dynamic model, the behavior of the pulse with the transition of 0-1-0 has been described. In this description, the transient pulse broadening of a node is computed as the difference between the propagation delays which is defined as  $t_{pLH}$  for an output transition from logical "0" to a logical "1" and  $t_{pHL}$  represents to a high to low output transition [26]. The propagation delay is measured between 50% of the transition points of the input and output waveform modeled by the following equation 3.4.

$$\begin{aligned} \Delta t_p &= t_{pLH} - t_{pHL} \\ \text{if}(t_{SET} > (k + 3)t_{pHL}) &\rightarrow t_{out} = t_{SET} + \Delta t_p \end{aligned} \quad (3.4)$$

Please note that the coefficients of the propagation delays of  $t_{pHL}$  and  $t_{pLH}$  are computed by a linear interpolation of the experimental characterization that we performed in [25]. It is known that each gates has different propagation behavior related to the difference between the propagation delays of  $t_{pHL}$  and  $t_{pLH}$  which the transient pulse broadening and filtering at a node is dependent to these values.

For performing the characterization, we used two modules: a signal generator and a scope. The signal generator is an Agilent 811101A-M2 330 MHz. This generator has been used in order to apply pulses at different frequencies and with different voltages. The scope is a LeCroy WaveRunner 44Xi model equipped with high-impedance calibrated probes and able to measure voltage transients larger than 200 ps with a time resolution of about 90 ps. The probes have been connected to the device under the evaluation to measure voltage and width of the generated transient pulses and the pulses that have been propagated through inverter logic gates.

For characterization of SET behavior, we developed an environment consisting of 5652 chains of inverters. The developed environment leads to observe the effect of the voltage glitch traveling from the injected point at the start of the chain through Flip-Flops. Considering the number of inverter gates (5652), the source SET duration and measured pulse duration at the end of the chain represented in Table 3.5. while  $t_{pHL}$  and  $t_{pLH}$  has been calculated for each SET pulse, represented in Table 3.4.

Please notice that the SET pulses have been selected in a way that it is possible to physically perform an electrical injection analysis. In order to inject SET pulses,

Table 3.4  $t_{pHL}$  and  $t_{pLH}$  for each Inverter

$t_{pHL}[\text{ns}]$	$t_{pLH}[\text{ns}]$
2.793	2.791
1.763	1.761
1.188	1.186
0.605	0.603
0.466	0.464
0.354	0.352
0.354	0.26575
0.214	0.216
0.192	0.198

Table 3.5 Propagation behavior

Input Pulse[ns]	Output Pulse[ns]
19.57	26.83
12.35	18.68
8.33	16.69
4.25	12.383
3.34	12.461
2.49	12.588
1.89	12.324
1.35	12.251
0.84	12.132

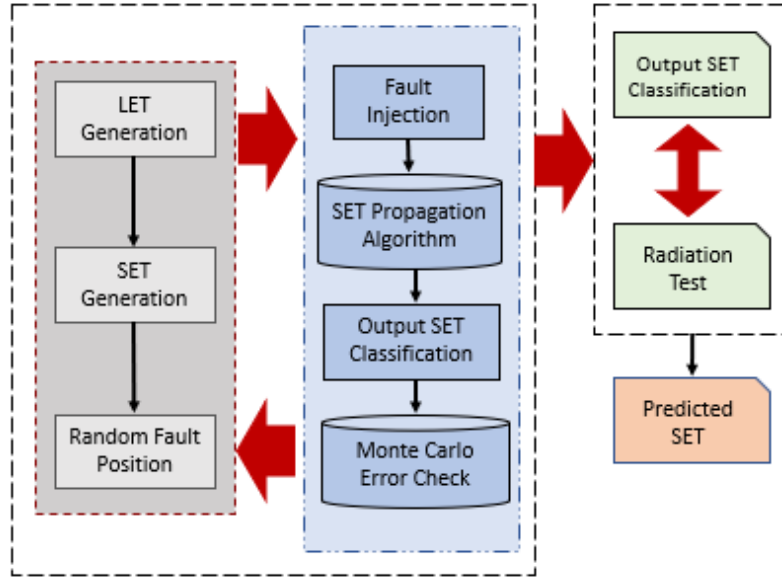


Fig. 3.28 The flow of the developed SET prediction method

a methodology based on internal electrical pulse injection has been used. This methodology leads to an accurate characterization of the SET propagation within the logic and routing resources of Flash-based FPGA [13]. The internal electrical pulse generator designed to create a SET pulse, leads to a better control of a SET parameters comparing to external injection. internal electrical injection has been explained in details in the following sections.

### 3.2.2 SET prediction methodology

The main core of the proposed approach is based on the prediction of the source SET in order to imitate the radiation particles strike condition with the random location and energy. Figure 3.28 represents the developed environment which consists of a Monte Carlo simulation block that is applicable to all kind of circuits implemented on Flash-based FPGA.

The flow starts with the generation of the SET pulse. Based on the LET, the SET generation model is creating an appropriate random SET pulse, applied to random location of Flash-based FPGA circuit. Fault injection phase is dedicated to inject generated SET pulses and propagate the generated SET through the circuit output

---

```

1: inv_tpHL= inv_gate_high2low_time
2: inv_tpLH= inv_gate_low2high_time
3: Function SET_Fault_Inj(SET):
4:     Foreach path startswith SET.position:
5:         SET_output=SET_Propagate(SET,path)
6:         SET_Classify(SET_output)
7: Function SET_Propagate(SET,path)
8:     SET_gate_in=SET
9:     Foreach gate in path:
10:    //Calculate the SET output for the gate based on SET analytical Model
11:    //Using inv_tpHL and inv_tpLH as parameters
12:        SET_gate_out= SET_gate_model(SET_gate_in, gate)
13:        SET_gate_in= SET_gate_out //Current output is input for next gate
14:    Return SET_gate_out
15: Function SET_Classify(SET)
16:     If not is_total_filtered(SET) Then:
17:         Report(SET);
18: While (Monte_Carlo_error > threshold) do
19:     F= the number of injections
20:     For I=1 to F Loop:
21:         SET_duration= random_SET_duration();
22:         SET_pos= random_SET_position();
23:         SET= gen_SET(SET_duration, SET_pos);
24:         SET_Fault_Inj(SET);
25:     End Loop
26: End While

```

---

Fig. 3.29 The flow of the developed SET prediction method

where the SETs are classified. As the next phase, the MonteCarlo flow has been applied to the environment.

MonteCarlo is a computational algorithm that is based on a repeated random sampling to obtain reliable numerical results. This methods is performed in three phases: The first phase is dedicated to generating simulation data. Following by performing some statistical procedures and recording the results. After repeating the results for multiple times and reaching to different results, the variability between the results has been calculated. This variability is called MonteCarlo Error (MCE), the extent to which difference occur across simulation depends on the setting on the MCE. The MonteCarlo executes until acquired MCE reaches the chosen by the user. Therefore, we developed a tool which the pseudo code has been shown in Figure 3.29.

The developed tool starts with the estimated propagation behavior of  $t_{pLH}$  and  $t_{pHL}$ . The flow continues with selecting randomly the SET pulse duration and location of injection. As a next phase, the tool continues with analysis of SET propagation through different paths of circuit with respect to SET analytical model and classifies the SET reached to the storage element. This classification has been

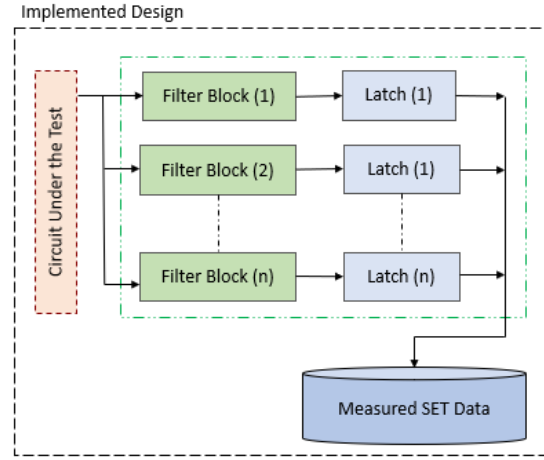


Fig. 3.30 Global scheme of the generated test setup.

compared to the experimental results and this comparison has been used to define MCE. This algorithm has been repeated to reach the selected value for the MCE.

### 3.2.3 Prediction of SET on Flash-based FPGAs

In order to confirm the proposed environment, a circuit consisting 5652 inverter chain has been designed and implemented on A3P250 ProASIC3 Flash-based FPGA. The experimental results are provided applying an empirical approach comparing the prediction model with radiation test campaigns data.

The radiation experiments have been performed at UCL facility of Louvain-la-neuve. The details of the performed experimental radiation test are reported in [25]. The developed benchmark has been under the beam for two LET values:  $Ag_{107}$  ion beam having  $LET\ 54.7\ MeV\frac{cm^2}{mg}$  and  $Ni_{58}$  ion beam with the LET of  $28.4\ MeV\frac{cm^2}{mg}$ . These tests have been performed on three different devices specifically provided by Microsemi. All the three devices have been preliminarily evaluated checking the correct operative functions at the testing frequencies of 50 and 100 MHz. The propagated SET pulses have been monitored at the output of the implemented setup. The logical scheme of the generated test setup is observed in Figure 3.30 while the scheme of the SET pulse measurement is represented in Figure 3.31.

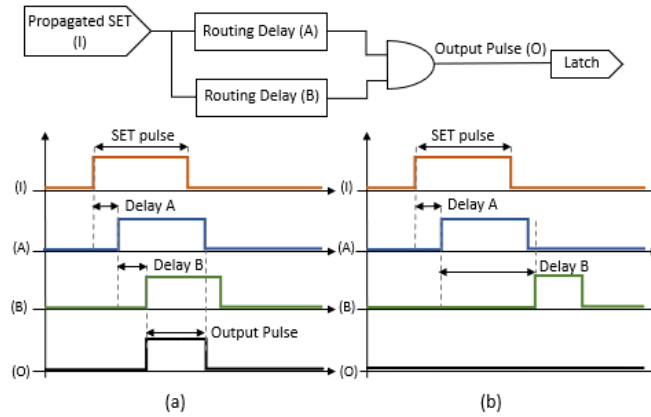


Fig. 3.31 Scheme of the SET pulse measurement circuit.

The SETs are categorized in three different groups related to the monitoring design system. The groups are coherent between the experiments as presented in Figure 3.32, 3.33 and 3.34.

We applied the MonteCarlo methodology to the benchmark circuits with three different standard error deviation values: 10%, 1% and 0.1%. The output SETs have been classified according to the intervals used to monitor SET during the radiation test experiments, represented in Figure 3.32, 3.33 and 3.34.

The results acquired by the proposed prediction model shows the SET pulses with the duration of 0.45 ns and 0.48 ns. This SET length predicted by the simulation model are closely matches with the radiation experiment result which leads to the verification for the proposed prediction methodology.

Considering the error-bar, all the classifications are comparable with an overall of about  $\pm 4.5\%$  with the MonteCarlo approach applied with a standard error deviation of 10%. Moreover, with respect to standard error deviation at 1%, the overall error is about  $\pm 0.2\%$ . Reducing to zero to match the prediction considering the standard error deviation of 0.1%, leading to a progressive precision of the proposed method.

### 3.2.4 Research advancement on the prediction of SETs

In order to perform an accurate analysis and mitigation of SET pulses, one of the golden steps is identifying the characteristics of the source generated SET pulse within the silicon structure of devices. The generated pulse is dependent on several

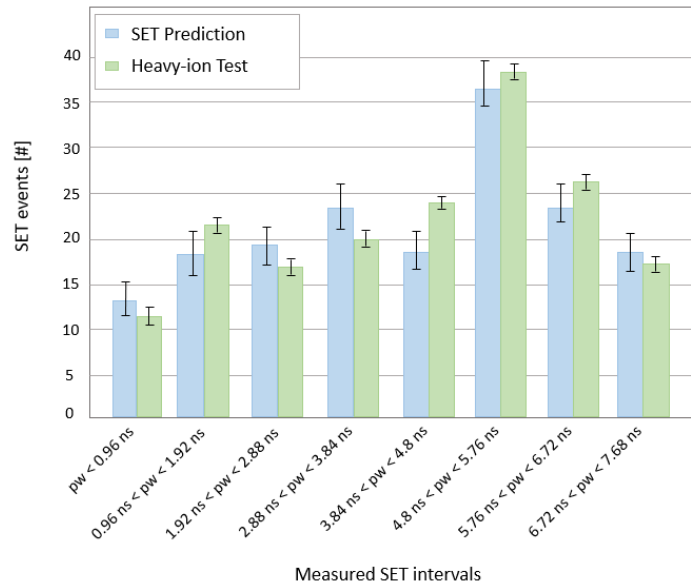


Fig. 3.32 Comparison between the SET prediction model and the heavy-ion test results. Standard error deviation at 10%.

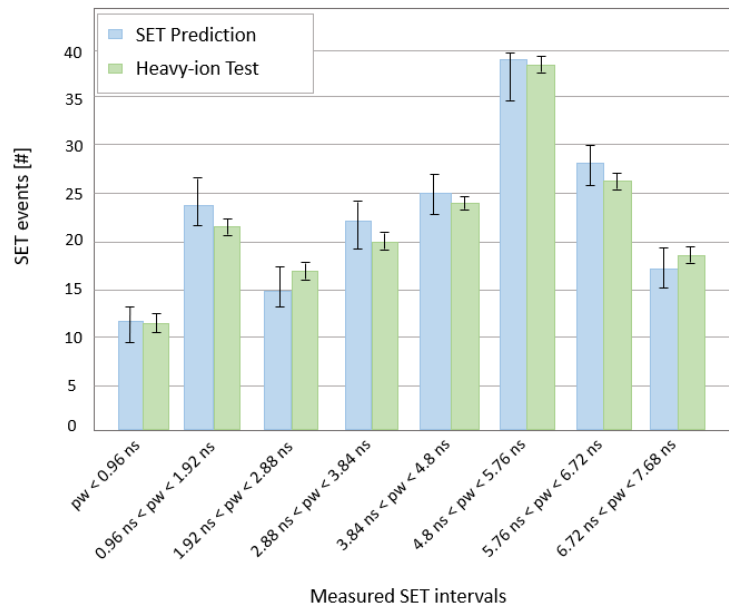


Fig. 3.33 Comparison between the SET prediction model and the heavy-ion test results. Standard error deviation at 1%.

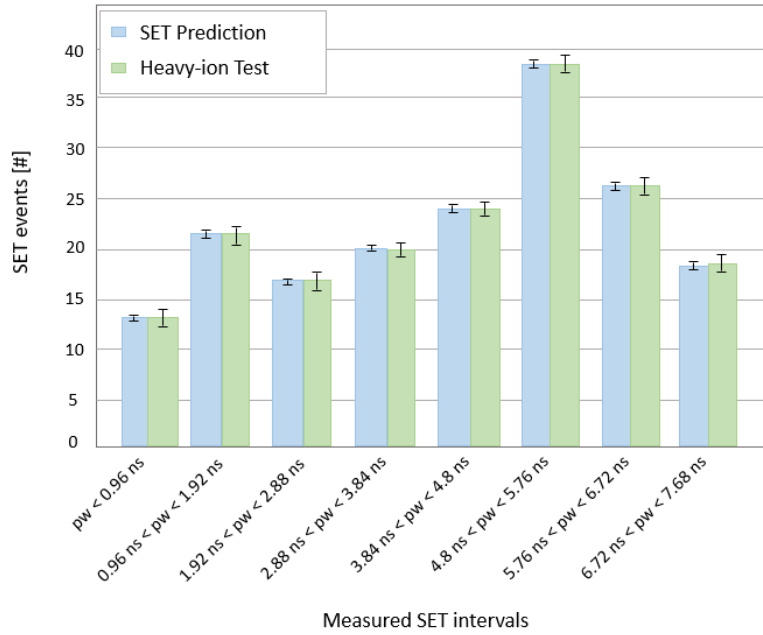


Fig. 3.34 Comparison between the SET prediction model and the heavy-ion test results. Standard error deviation at 0.1%.

parameters such as the radiation profile, the type of the existing particle, incident angel and the technology under the study. Therefore, a model has been developed which takes into account the mentioned parameters and provide the characteristics and the duration of the expected pulse.

As a conclusion, a novel Single Event Transient prediction model has been proposed which leads to the effective identification of SET phenomena and correlates the radiation particle energy and the type of transient effect. With respect to the experimental result acquired from heavy-ion particles, the proposed method has been validated.

### 3.3 Single Event Transient Analyzer - SETA

After the generation of SET pulse, the generated pulse may propagate through the routing and logic resources of the circuit that it may reach and be captured by a storage element such as Flip-Flops(FFs). The SET sampled by a FFs may corrupt the previously stored values, causing a bit-flip which in turn can propagate and leads to misbehavior of the system. Therefore, it is mandatory to evaluate and analysis



the SET phenomena [27]. Several studies have been dedicated to the evaluation of SET propagation using electrical simulation [28]. Even though these methods are effective for analyzing the propagation of SET pulses, they do not evaluate the broadening or filtering effect of the SET pulses traversing logic and routing resources knowing as Pulse Induced Propagation Broadening (PIPB) [29]. In addition, those approaches are time consuming. Therefore, not sufficient enough to be applied to an industrial design flow with enormous amount of resources. Among the studies available using FPGA technologies, FPGAs with Flash-based configuration cells are mainly addressed since their configuration memory cells are essentially immune to bit-flips. Therefore, SETs in the Flash-based FPGAs logic and routing resources are the major concern regarding soft errors. Many studies have focused on identifying the type of generated SET within the silicon structure of Flash-based FPGA with respect to the SET behavior of sequential and combination circuits [20] [30] [31]. This evaluation of SET effect on the logic and routing structure are used for verification of the efficiency of the SET mitigation based on electrical filtering [32]. Radiation test experiment and electrical fault injection of SET propagation of custom circuits designed specifically to observe SETs have been reported [31].

### 3.3.1 SET behavior in combinational logic

In order to analyze the sensitivity of a FPGA regarding SET event, an elaboration of FPGA architecture is required which is based on the behavior of SET pulse propagating through routing and logics of the implemented design. Therefore, an accurate modeling of SET phenomena induced by radiation particles within the silicon structure of the nanometer devices is required. During this modelization, one of the main goal is to focus on the PIPB effect. Please notice that SPICE is not able to simulate the broadening or filtering effect of the pulse propagated through the logic. Therefore, Matlab environment is used to provide the physical evaluation of this effect. To do so, several features of the design are required such as: Technology information, thickness, area, resistance and capacitance of interconnection and device layers.

As a primary phase of SET model, according to the characterization provided in [33], SET pulse is generated. This formulation is reported in equation 3.5 where

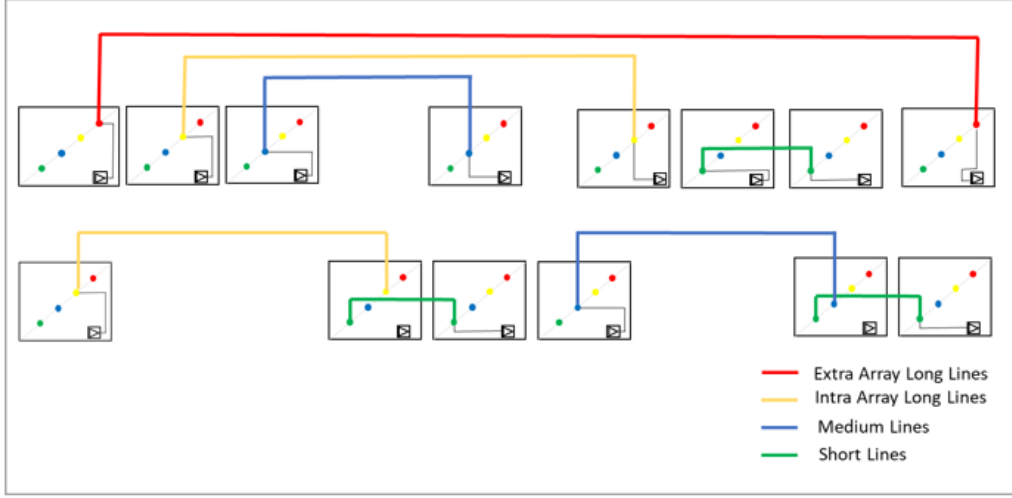


Fig. 3.35 The device routing topology of the Microsemi ProASIC3 family.

$\tau_n$  is the duration of the transient pulse .

$$\begin{aligned}
 &1. \text{if}(\tau_n < k\tau_p) \rightarrow \tau_{n+1} = 0 \\
 &2. \text{if}(\tau_n > (k+3)t_p) \rightarrow \tau_{n+1} = \tau_n + \Delta t_p \\
 &3. \text{if}((k+1)t_p < \tau_n < (k+3)t_p) \rightarrow \tau_{n+1} = \frac{\tau_n^2 - r_p^2}{\tau_n} \\
 &4. \text{if}(kt_p < \tau_n < (k+1)t_p) \rightarrow \tau_{n+1} = (k+1)t_p(1 - e^{-(\tau_n/\tau_p)}) + \Delta t_p
 \end{aligned} \tag{3.5}$$

This mathematical model has been developed by Matlab environment in order to model the generated SET pulse propagating through the logic gates.

### 3.3.2 SET behavior in routing interconnections

The developed routing model is mainly based on an accurate calculation of the propagation delay of the routing system. In order to do this, the routing segments has been classified in four groups: extra array long line which are the longest interconnections, intra array long lines for long connections through the whole device, medium lines and short lines for local routing resources. Figure 3.35, represents routing structures.

For considering the effect of the routing on the whole circuit, we extracted the coordinate of the logic functions. Using these coordinates, it is possible to know the

Table 3.6 Routing topology organization on ProASIC3 devices

Kind	Delay[ns]	Distance[#]
Short	0.8	1
Medium	1.01	2
Intra long	1.265	6
Extra long	1.239	12

number and kind of segments used between two logic functions. By calculating the propagation delay of routing segments, we are able to assign the propagation delay to the related kind of routing. In Table 3.6, the data obtained from the ProASIC3 Microsemi device family. Considering the routing structure in this family, it is possible to analyze the routing effect on SET propagation on the whole device.

### 3.3.3 Integration of SETA with commercial tools

In order to evaluate the effect of SET on the functionality of the circuit implemented on Flash-based FPGA, we developed a CAD methodology which is able to evaluate not only the propagation of the SET pulse, but also the possible pulse propagation caused and the PIPB effects on industrial circuit with enormous resource usage implemented on Flash-based FPGAs [34]. The developed CAD tool, named *SETA* is interfacing with the commercial FPGA design flow providing an SET sensitivity analysis of the implemented design.

To elaborate more, the goal of the proposed CAD tool is to provide an effective methodology for analysis of SET sensitivity taking into account different SET propagation scenarios. *SETA* is integrated with the standard FPGA design flow, as the flow is illustrated in Figure 3.36.

The developed tool is starting from the hardware description of the design (HDL), going through netlist synthesis, mapping and place and route. Using the commercial tool, we generate the post-layout netlist along with the Physical Design Constraint (PDC) file which consists the information regarding placement of the implemented design. The developed *SETA* tool starts from elaborating the post-layout netlist and Physical Design Constraint file of the design to perform the SET analysis. For performing the SET analysis, *SETA* evaluates the pulse propagation behavior through the routing and logic resources of the design in order to generate the SET

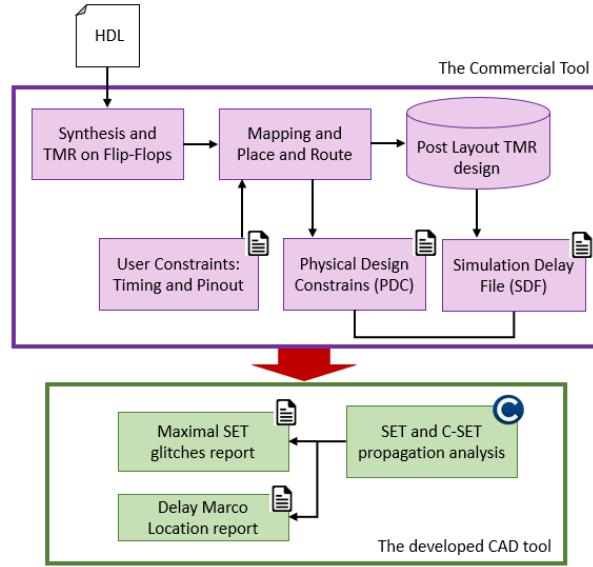


Fig. 3.36 The developed analysis flow for the accurate evaluation of SET effects on SoC implemented on Flash-based FPGA.

sensitivity. To elaborate more, *SETA* reports all Gate-to-Gate PIPB coefficient during the propagation of SET pulses.

### SET Analyzer

To evaluate the SET sensitivity of the circuit implemented on Flash-based FPGA, *SETA* works in two phases: 1. generation of SET pulses 2. elaborating the physical description of the design.

As a first phase, *SETA* generates the SET pulses and inject the generated SET pulses in all the logic resources of the circuit. In order to do this, user should set the parameters such as voltage amplitude and pulse width. Based on this parameters, *SETA* generates a list of SET pulses. Each pulses is described using 100,000 voltage sample point, allowing a precision of 1 ps.

After the generation of SET pulse, *SETA* starts elaborating the PDC file containing placement information of the logic resources used in the circuit. This information together with the logic information extracted from from post-layout netlist have been used to generate a Physical Design Description (PDD) file for extracting and storing the elaborated circuit where I/O pins, Flip-Flops, RAM or ROM ports are considered as terminal nodes connected through routing segments.

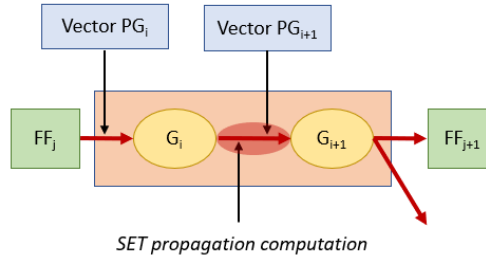


Fig. 3.37 The SET propagation approach: a voltage vector  $PG_i$  is transformed in a new vector  $PG_{i+1}$ , considering the gate  $G_i$ , the subsequent gate  $G_{i+1}$  and the routing segment between two logic gates.

As it has been mentioned before, there are four different types of routing segments: Extra array long lines; intra array long lines; medium lines and short lines [30].

The generated SET pulses of the first phase are injected at each single intermediate node of the design and propagated until a terminal node. During this propagation, based on the type of gates and routing interconnection stored in PDD file, PIPB coefficient is calculated. At the end, *SETA* tool reports the SET sensitivity for each terminal node, in terms of final PIPB coefficient and probability of SET causing bit-flip.

Considering the calculation of PIPB, *SETA* describes the voltage glitch as an array  $P$  of real types values. Considering a scenario where a voltage glitch is propagated from the logic element  $G_i$  to a subsequent logic element  $G_{i+1}$ , the *SETA* tool uses an array  $PG_i$  to describe the voltage transition at the input of the gate  $G_i$  and an array  $PG_{i+1}$  to describe the new voltage array which is generated after the propagation from  $G_i$  to  $G_{i+1}$ . This concept is explained in Figure 3.37.

The computation of the SET pulse propagation is the golden feature of the proposed algorithm. During this computation, a set of parameters were stored in a device library including the port delay, the routing delay and the Manhattan distance of each pair of source and destination gate. The mentioned delay is calculated with respect to the resistive and capacitive load for each VersaTile input as sum of the fan-in and fan-out contributions of the logic cells. These parameters are described as PIPB factor coefficient represented in Figure 3.38. The algorithm is following some steps for calculating the PIPB factor:

1. Given the pulse width  $PW_1$  at the source gate( $G_i$ )

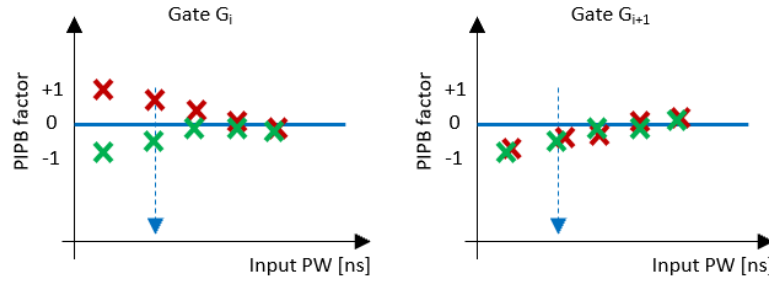


Fig. 3.38 The main core of the Propagation Induced Pulse Broadening (PIPB) calculation for the involved gates gate  $G_i$  and the following gate  $G_{i+1}$  (red X is representing the worst condition while green x is representing the best condition).

2. Identification of the destination( $G_{i+1}$ )
3. Calculation of the sum between the best and the worst propagation condition defined as  $PIPB_{BEST}$  and  $PIPB_{WORST}$  of the two considered gates  $G_i$  and  $G_{i+1}$ .

Equation 3.6 is representing the mentioned calculation. The best and worst conditions are corresponding to the timing behavior of the gate couple calculated respectively as follows: best condition under low temperature, high voltage and fast process corner; worst case under high temperature, low voltage and slow process corner. For both of the gates, the PIPB factor have been calculated. Each PIPB is obtained by interpolating the gate PIPB characteristics with the input pulse width calculated based on the input pulse vector  $PG_i$ . The final  $\Sigma PIPB$  used to reshape the pulse described by the output pulse vector  $PG_{i+1}$  is the sum of contributions of two gates.

$$\begin{aligned} PIPB_{BEST} &= PIPB_{BEST-G_i} + PIPB_{BEST-G_{i+1}} \\ PIPB_{WORST} &= PIPB_{WROST-G_i} + PIPB_{WORST-G_{i+1}} \end{aligned} \quad (3.6)$$

In order to identify the effective behavior of SET pulse broadening or filtering while propagating through the routing and logics, we imitate the plot of the PIPB coefficient in Figure 3.39. Please note that the worst and best conditions are described by specular behavior of the  $\Sigma PIPB$ . Moreover, the decreasing or increasing of the wire length which is defined as number of wire segments between the two considered gates is acting strongly on the values of the PIPB but not on the intrinsic gate BIBP behavior. Indeed, an extremely large or low number of wire length saturates the PIPB

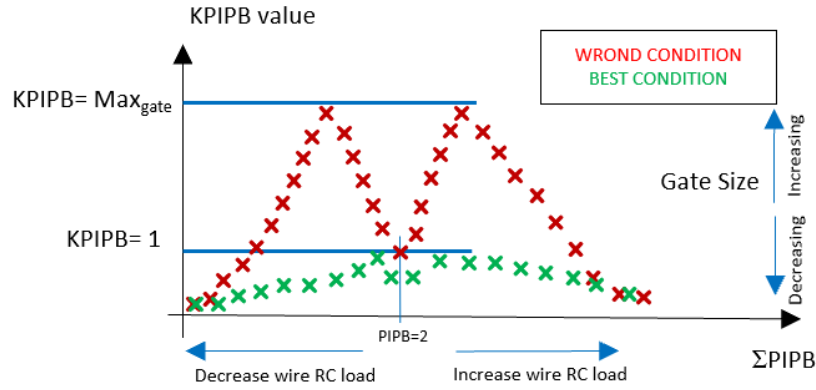


Fig. 3.39 Representation of the cumulative (KPIPB) on a generic couple of gates considered in a pulse traversing computation (red X is representing the worst condition while green x is representing the best condition).

Table 3.7 Characteristic of the original benchmark circuits

Circuits	Versatile[#]	FFs[#]	Frequency[MHz]
B05	415	66	47
B09	493	67	46
B12	565	123	48
B13	162	50	52
CORDIC	956	240	45
RISC	1,401	1,156	42

characteristics. Moreover, as it can be observed, the size of the versatile, specially the number of internal resources programmed by the versatile logic element is affecting the final PIPB effect.

### 3.3.4 SETA on Flash-based FPGAs

The proposed environment has been applied to circuits implemented on A3P250 Flash-based FPGA manufacture by Microsemi with 6,144 logic versatile. We select circuits with different complexity: four circuits from the ITC99 benchmark collection [22], a Cordic core and a RISC microprocessor. The characteristics of the circuits are reported in Table 3.7. For each circuit, the number of Versatile used as logic function or Flip-Flop and the maximum working frequency is reported.

Table 3.8 Sensitivity report of the selected circuits for injection of 5000 SET pulses lower than 1 ns

Circuits	logical masked[#]	filtered[#]	partially filtered[#]	broadened[#]
B05	46	9	3	8
B09	47	3	6	11
B12	102	1	7	13
B13	21	14	8	7
CORDIC	161	12	28	39
RISC	572	204	184	196

We analyzed the circuits SET sensitivity using the developed tool. Microsemi Libero SoC commercial design tool has been used for generating the inputs required for the developed environment such as PDC and post layout netlist. For our experiment, we consider SET sources as SET pulses with the duration less than 1ns (0.3, 0.6 and 0.8 ns) since these SET pulses are known as the most probable events generated by heavy ions striking the Flash-based FPGA 130nm technology [35].

As an output of the developed tool(*SETA*), the condition of each Flip-Flop with respect to the propagated SET pulse is reported:

- *Logical masked* is representing the Flip-Flops which are facing the SET pulses that have been filtered by circuit logical masking.
- *Filtered* is dedicated to the Flip-Flops which are facing the pulses that are electrically filtered during their propagation.
- *Partially filtered* is the case of the Flip-Flops that are facing the SET pulses which the duration of the pulses decreased before reaching to the Flip-Flops.
- *Broadened* is representing the case of the Flip-Flops which are facing pulses with broadened duration.

Table 3.8 is reporting the sensitivity analysis of the mentioned circuits using *SETA* tool while we inject 5,000 SETs for each circuit.



### 3.4 Evaluation of Transient Errors in GPGPUs

General Purpose Graphic Processing Units (GPGPUs) are high-performance oriented devices designed to execute stream processing computations providing high computational power combined with an overall low design cost thanks to their flexible development platform. The parallelism capabilities of GPGPUs leads them to be a suitable option to be adopted in mission-critical applications including automotive, avionics, space and bio-medical fields [36]. For example advanced driver assistance system (ADAS) applied to cars and it is largely based on the usage of images or radar signals coming from external camera and sensor devices to detect possible obstacles triggering the breaking system. On the other side, considering the space application, European Space agency (ESA) is applying low power GPGPUs for images compression on the COROT satellite [37]. Therefore, the required bandwidth for sending the data is minimized. Moreover, the Airbus avionic company within the framework of the ARAMIS project integrates all the electronic used to deploy the Collision Avoidance System (CAD) into a single board including a GPGPU core [38]. High degree of parallelism as the main feature of GPGPUs, makes it highly easy to implement in software transition soft error mitigation methods such as Duplication With Comparison (DWC) and Triple Modular Redundancy (TMR). However, the dimension of these software, their component organization and their complexity make them sensible to soft errors [39] [40]. Moreover, while many soft-errors hardening methodologies already exist for system based on traditional CPUs, solutions and evaluations for GPGPUs are still under investigation and development [41].

Different methods such as fault injection is adopted for evaluating the transient errors sensitivity of GPGPUs. Fault injection is a commonly adopted solution for validating the final application code and check its detection and correction with respect to transient errors. Fault injections methods can be categorized in two groups: The first one is to expose the GPGPUs to accelerated radiation beams. The second is resorting to transient errors injection using a simulation-based software method. One of the main restriction of the developed method is elevated economical costs of experimental radiation beam and the excessive intrusiveness combined with ineffective fault models of simulation-based approaches.

Preliminary activities have been performed through radiation test experiments where the GPGPUs operation is executed with the device being irradiated by a neutron radiation source with energy above 10 MeV. Neutron particles may generate

transient errors which cause silent faults or functional interrupts. These methods are applied to test software-based hardening strategies avoiding the propagation of soft-errors [41]. Several studies present the application of the Error Correction Code (ECC) mechanism in the most common applications in the High Performance Computing (HPC) and safety-critical domain [42] [43].

Fault injection by emulation has been recently developed in [44] [45]. These methods are based on the NVIDIA CUDA debugger(gdb) which by means of ad-hoc software infrastructure is injecting transient faults in the accessible memory components to mimic faults affecting ALUs and FPUs and classifying their effects.

These methodologies are characterized by the advantage of using a real device for transient error characterization in order to evaluate the behavior of transient errors and its realistic propagation behavior within the physical architecture of GPGPU. However, it has two main disadvantages: firstly, the computational speed of GPGPI under the test is really low due to the debugger interface. Therefore, it nullifies the benefit of the emulation; secondly, debugger interfaces have limited resource accessibility. Therefore, the fault injection is limited only on specific variables and memory elements of the architecture without any possibility of emulating transient effects affecting GPGPU combinational resources.

Many GPGPU simulators are developed based on the Instruction Set Architecture (ISA) and hardware architecture model. [46] represents the Barra GPGPU simulator which is based on UNISIM framework and allows the Parallelization of the Thread eXecution (PTX) and the execution simulation of CUDA programs at the functional level. Moreover, the simulator can be customized reusing the module libraries and feature proposed in the UNISM repository and it allows the integration into the NVISIA OpenCL software stack.

The performance of GPU simulation has been evaluated in [47] by simulating NVIDIA parallel thread execution with non-graphics applications, decreased threads and characterizing the performance differences when different DRAM locality algorithm are adopted. In [48], a new simulator, GPGPU-sim, is developed which uses different benchmark application [49] in order to explore efficient mechanism in Single-Instruction Multiple-Data (SIMD) branch execution on GPUs. Using GPGPU-sim, users are allowed to have a detailed model of the commercial GPGPU such as Fermi and FT200. The model includes the specification of the streaming-multiprocessor architecture, the existence and the size of the caches level L1 and

L2 as well as the shared memory architecture and size. The architectural simulator GPGPU-sim offers the flexibility to modify the processor parameters and allows the integration of customized modules.

Moreover, the GPGPU-sim has been used in occurrence to hardware emulation for the execution of fault-injection analysis [50]. However, this method has the limitation of analyzing transient errors affecting exclusively variables and registers used by the tested applications without the possibility to analyze the effects of faults generated in the internal logic structure [51].

We propose a simulation-based fault injection methods which provides solutions for disadvantages of the state-of-the-art and providing novel insight on the behavior of GPGPUs when affected by transient errors. In this simulation environment, we inject transient errors within the gate level model of a GPGPU device and evaluate its consequences on the executed application. Please notice that using this environment, as accurate GPGPU fault model is provided which allows to propagate transient errors from the affected location to the registers involved in the computation, modeling the transient error propagation. Therefore, it is possible to determine the right influence of the error in the GPGPU architecture. Moreover, the low intrusiveness of the proposed approach can be mentioned specially when a software application is evaluated. During this analysis, Streaming Processors (SPs) are the main focus. These units are the elementary units where arithmetic and logic computations are executed. Therefore, a SP is one of the most critical components of GPGPUs computing because an error may affect the execution code, the operands or the results thus compromising the whole algorithm execution.

### 3.4.1 The proposed environment

The proposed environment is represented in Figure 3.40. The first steps of applying this method is availability of the hardware model (VHDL) and the Instruction Set Architecture (ISA) of the considered GPGPU. The fault injection method includes two steps: The first step is the injection of the transient errors while the second is the GPGPU application simulation execution with the injected fault. As the first step, transient errors is injected in the hardware architecture model of the GPGPU under test. For generating the transient error, we considered the worst case scenario. It means that SET pulses are generated without considering the logical and timing

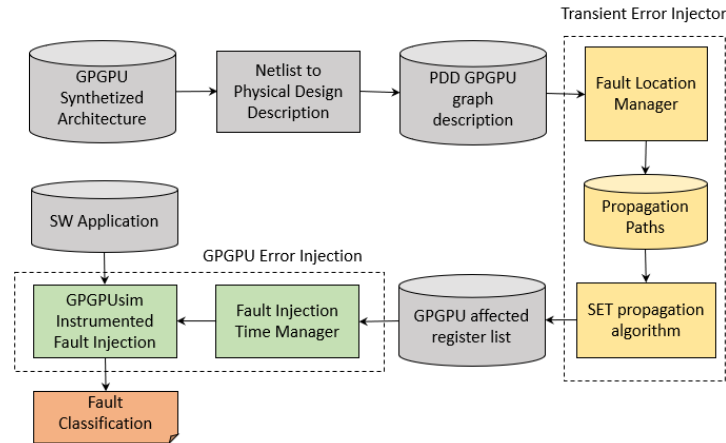


Fig. 3.40 The flow of the developed simulation-based fault injection for transient errors analysis on GPGPUs.

masking. After the generation of the pulse, the pulse is propagated and stored in the GPGPU affected register list. The second phase goes on with the execution of the hardware application using GPGPU-sim framework, which has been instrumented with the injection of errors that simulates the radiation effects in a realistic way. The injection has been performed with respect to the timing analysis and clock cycle. To elaborate more, GPGPU affected register list is used to perform the error injection and simulating the effective propagation of the original transient pulse to the GPGPU computation. At the end, the result of each fault injection is classified.

### Architecture level transient error injection

The main goal of architectural level transient error is to simulate the Single Event Transient (SET) or transient error phenomena generated by radiation particles interacting within the silicon structure of the GPGPU device. Therefore, a description of GPGPU structure is required. To do so, as a preliminary phase, extracting the architectural graph description is performed. A software tool is elaborating the GPGPU netlist and translating it into a Physical Design Description (PDD) file containing a directed graph representation of the circuit where each vertex models a logic gate or sequential element while edges model the interconnection between them.

The transient error injection module is performed in the following steps: the generation of the SET pulse modeling as a transient pulse shape, the localization of the combinational gates within the circuit description and execution of the propagation

of the SET pulse starting from the selected sensitive node of the GPGPU circuit and traversing logic gates and routing interconnections until a storage element.

For generating the SET pulse, we developed a model for elaborating the physical layout description. The first phase generates the SET model with respect to the definition represented in Equation 3.7. The second phase is dedicated to the propagation with respect to the basis of the resistive and capacitive load calculated on the hardware technology model of the circuit. The propagation coefficient is used in the model described in Equation 3.7 in order to generate the expected propagation coefficients for all the logic paths. Due to the difference between propagation delays of  $t_{pHL}$  and  $t_{pLH}$ , the propagation behavior of SET pulse is defined.  $t_{pLH}$  stands for an output transition from logical 0 to logical 1.  $t_{pHL}$  refers to high to low output transition and  $\Delta t_p$  represents the difference between them.  $T_n$  represents the duration of the transient pulse at the  $n$ th logic state and  $t_p$  is equal to the  $t_{pHL}$  for a one to zero to one transition at the  $n$ th node or equal to  $t_{pLH}$  for a one to zero transition and  $k$  is a filtering parameter which depends on the technology of interest.

$$\begin{aligned}
&1. \text{if}(\tau_n < k\tau_p) \rightarrow \tau_{n+1} = 0 \\
&2. \text{if}(\tau_n > (k+3)t_p) \rightarrow \tau_{n+1} = \tau_n + \Delta t_p \\
&3. \text{if}((k+1)t_p < \tau_n < (k+3)t_p) \rightarrow \tau_{n+1} = \frac{\tau_n^2 - t_p^2}{\tau_n} \\
&4. \text{if}(kt_p < \tau_n < (k+1)t_p) \rightarrow \tau_{n+1} = (k+1)t_p(1 - e^{-(\tau_n/\tau_p)}) + \Delta t_p
\end{aligned} \tag{3.7}$$

The architectural level transient error injection is executed for all the desired number of injected SETs. The generated outcome consists in a database of SET pulse events observed in all the GPGPU registers that will be selected for simulation-based fault injection. The golden benefit of transient injector module is individuating the effective propagation of the SET on the computational registers which leads to a more accurate fault injection execution of the GPGPU soft error simulation.

### **GPGPU soft error simulator injection**

The main advantages of architectural simulator is providing a robust solution for verification of the efficiency and application performance through the detailed models of the most used commercial devices. In our work, we used the GPGPU-sim [45] due to the flexibility for developing a soft error injection that simulates the radiation

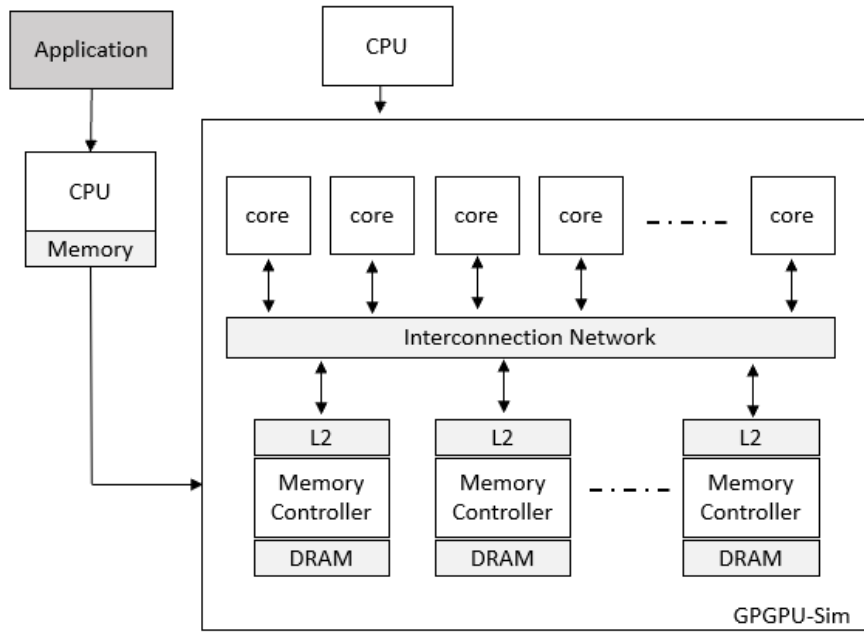


Fig. 3.41 GPGPU-sim modeled system.

effect realistically. The overall model of the GPGPU architecture is represented in Figure 3.41.

We modified the GPGPU-sim simulator to perform dynamic fault insertion in the thread executed by the GPGPU. Threads are scheduled in the groups of maximum 32 threads known as warps. Simulation executes GPGPU kernel and warps sequentially. By accessing the global memory, it is possible to extract the information related to the execution timing. Due to the thread class interface, we can obtain all the information to identify each thread execution in the simulator and the hardware associated to it. A fault-free execution allows the extraction of the execution timing information of an application. Therefore, we can know all the instructions executed by the simulator and identify the correspondent instruction registers and computational operands registers.

Figure 3.42 represents the scheme of the soft-error injection tool integration in the GPGPU-sim.

We implement a fault injector module for injecting a specific fault inside an executed instruction. This module runs the applications of two distinct operational modes: 1. Golden response mode: refers to fault-free execution. Therefore, at the

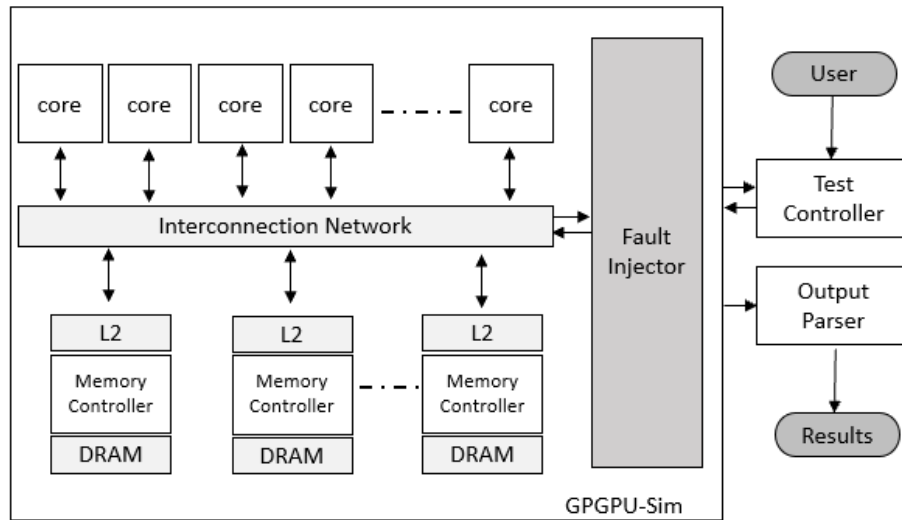


Fig. 3.42 Soft-error injection tool integration in the GPGPU-sim simulator.

output, it provides a profiling file including a list of all the instructions executed with information of the warp and thread in a specific execution time. 2. Fault simulation mode: for injecting a fault inside an instruction. For every fault generated a description of the fault is obtained.

The fault injection module operations can be managed through the usage of a configuration file. It allows to enable the fault injection module. Consequently, using the fault list, it selects the instruction in which the injection has to be performed. Every fault is identified from the following parameters:

1. Instruction name: the type of instruction affected from the injection.
2. Warp identification: identify if the fault affects a specific warp or all the warps.
3. Kernel identification: Identify if the fault affects a specific kernel or all the kernels.
4. Bit mask: it is a 32-bit integer representing the mask applied to the operands in order to invoke bit-flip. Each not-masked bit correspond to the relative injected soft-errors.

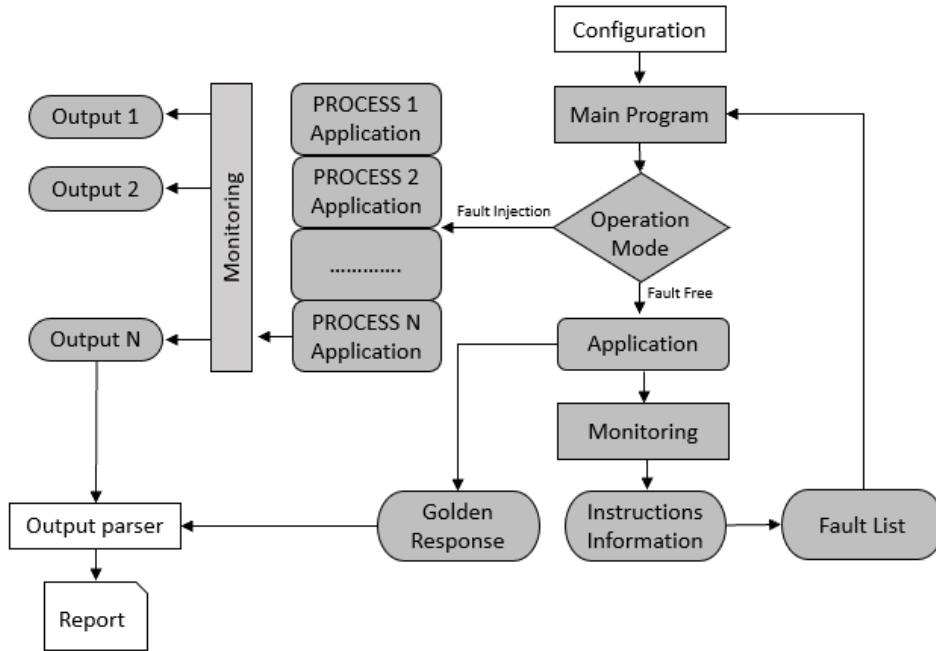


Fig. 3.43 Soft-error injection tool integration in the GPGPU-sim simulator.

### Test execution flow

The complete flow of the test execution is represented in Figure 3.43. The fault injection environment is mainly based on the integration of the architectural level transient injector and the GPGPU soft error simulator. The number of injected transient faults together with the number of parallel processes should be defined by the user. The injection rate is calculated with respect to the GPGPU affected register list automatically. Obviously, the simulation duration is dependent directly to the number of required simulation and the complexity of the application. As a first step of the test execution, compilation and building the application on the simulator is performed. Once this step is done, the golden response mode is set and a first execution of the application is performed. As a result, the execution timing data of the kernel and warps are extracted which leads to the complete set of instructions and golden response of the applications and is used to create an accurate timing fault list.

While the main program runs the application, the fault injection is performed and the injector module is checking the execution continuously in order to inject the error in the instructions and register defined in the GPGPU affected register list.



Once the injection time is gained, the GPGPU affected register list is used to identify which bit within the affected registers are flipped. As the end of the simulation, a log file is generated. In this log file, the executed instructions during the application, the information on the faults injected and the final results are reported. As an output, there is a report which is providing the number of affected simulation which can be classified as follows:

1. Silent simulation: defined as simulations that have not been affected by the fault injection and the results are equal to the golden one.
2. Timed out simulations: simulation which face a functional interruption due to the time threshold.
3. Corrupted simulations: simulations without reporting a functional interruption. However, the results are different from the expected one.

### 3.4.2 Fault tolerance design methods on GPGPU

Time redundancy and space redundancy techniques are the common methods applied to safety critical applications. In a case of a GPGPU, when the software code is implementing a function, the function can be executed twice and then results are compared. To elaborate more, the functions can be executed twice in two different threads on two different cores, having a form of Duplication With Comparison (DWC) in space. Applications can also execute the function three times on three different cores and choose the result with a voting technique which is known as TMR. However, these methods are not applicable to the faults in the control units that may cause the function interruption. However, the main goal is to apply concurrent execution of the CUDA kernels or CUDA functions on different CUDA cores.

#### Matrix product: plain and fault tolerance algorithm

The pseudo-code of the developed algorithm for matrix multiplication is shown in Figure 3.44. The sizes of the matrices A, B and C are not mentioned in the algorithm. However, respectively, they are  $m*n$ ,  $n*p$ ,  $m*p$ . Routines at rows 6 through 10 execute the block transfer from the matrices A and B allocated in the device RAM to the shared memory. In this phase, every thread only copies one element of the matrix

A and one element of matrix B, in case indexes are within the range of the matrices. Otherwise, it only initialized the corresponding elements in the share memory to zero. On the other hand, from row 21 to 24 the multiplication is executed to compute the element corresponding to the thread index. The value resulting from this operation is accumulated in the variable *valC*. At the end, from row 27 to 30, the end of the block iteration and the assignment of the final value to matrix C is reported.

To elaborate more, the number of accesses to an element  $A[i,j]$  or  $B[i,j]$  from the GPU main memory is equivalent to  $[m/b]$  and  $[p/b]$ , because every block of A is accessed every time a C result block is computed lying in the same row, while every block B is accessed every time a C result block is computed lying on the same column. The number of accesses in the share memory, per element, is equal to  $[m/b]*b$  for the elements of A and  $[p/b]*b$  for the elements of B because we access the elements  $b$  times within the block multiplications, and there are  $[m/b]$  block multiplications involving every block of A and  $[p/b]$  block multiplications involving every block of B.

The developed matrix algorithm requires some extra access with respect to the version without dividing the multiplication in blocks and using the shared memory. However, comparing to the computing of the product directly, the implementation offers a time advantages. The algorithm was executed on matrices size of  $n*1024*1024$  starting from  $n=1$  and doubling it every time, considering a running frequency of 1058MHz with a memory transfer rate of 2500MHz, it is possible to get a drastic improvement on performance as it is presented in Figure 3.45. However, as the main advantages of this method it can be mentioned that it is limited to multiply matrices whose result C has a number of elements not greater than maximum number of threads instantiated on the GPGPUs, as a thread only computes one element of the C matrix. Although, it is normally enough for most of the real world problems.

We choose three different matrix product algorithms: Duplication With Comparison (DWC), Tripe Modular Redundancy (TMR) and Algorithm Based Fault Tolerance (ABFT).

The DWC implementation is executed twice on two distinct output buffers as explained in algorithm represented in Figure 3.44. A second kernel compares the two results and if they do not match, an error is detected. In one way, the result is recomputed one time on a third output buffer and the value which appears two times are returned. Therefore, we can have three buffers allocated in case of error.

---

```

1: procedure MATMULKERNERL(matrix A, matrix B, matrix C)
2:  $i_a \leftarrow \text{bid}_y.b + \text{tid}_y$ 
3:  $j_b \leftarrow \text{bid}_x.b + \text{tid}_x$ 
4:  $\text{valC} \leftarrow 0$ 
5:  $i \leftarrow 0$ 
6: while  $i < [n/b]$  do:
7:    $j_a \leftarrow i.b + \text{tid}_x$ 
8:    $i_b \leftarrow i.b + \text{tid}_y$ 
9:    $\text{valA} \leftarrow 0$ 
10:   $\text{valB} \leftarrow 0$ 
11:  if  $i_a < m \ \& \ j_a < n$  then:
12:     $\text{valA} \leftarrow A[i_a, j_a]$ 
13:  end if
14:  if  $i_b < n \ \& \ j_b < p$  then:
15:     $\text{valB} \leftarrow B[i_b, j_b]$ 
16:  end if
17:   $A_{\text{shared}}[\text{tid}_y, \text{tid}_x] \leftarrow \text{valA}$ 
18:   $B_{\text{shared}}[\text{tid}_y, \text{tid}_x] \leftarrow \text{valB}$ 
19:  Synchronize threads
20:   $j \leftarrow 0$ 
21:  while  $j < b$  do:
22:     $\text{valC} \leftarrow \text{valC} + A_{\text{shared}}[\text{tid}_y, j] \cdot B_{\text{shared}}[j, \text{tid}_x]$ 
23:     $j \leftarrow j+1$ 
24:  end while
25:  synchronize thread
26:   $i \leftarrow i+1$ 
27: end while
28: if  $i_a < m \ \& \ j_b < p$  then:
29:   $C[i_a, j_b] \leftarrow \text{valC}$ 
30: end if
31: end procedure

```

---

Fig. 3.44 The matrix multiplication kernel algorithm.

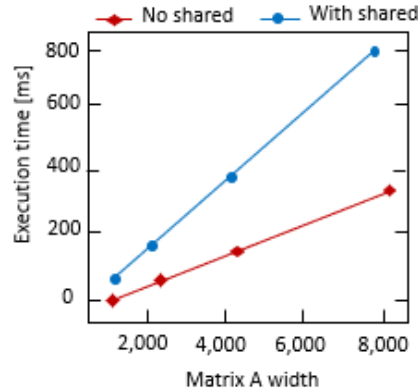


Fig. 3.45 Matrix product implementation comparison with and without using the shared memory. The developed algorithm is using the shared memory thus getting an improvement in performances.

In addition, it is necessary to copy the right result from its buffer to the final output buffer if they are different. In the development solution, we recomputed the product two times until they agree. In such a way, we use two buffers for the output instead of three, but it is possible to have a bigger time penalty in case of error. Figure 3.46 is representing the pseudo code of the algorithm developed for DWC. The two calls to MATMUL of row 6 and 7 can be executed on different CUDA cores, so that, if the hardware has enough resources enables, they can be executed in parallel. Despite of the simplicity of the DWC solution, it is expensive in terms of memory and time, especially in case of error.

Another method to handle faults is the modulation of the computation and check for errors at module level. Therefore, the algorithm recomputes only the corrupted module. To elaborate more, based on the Triple Modular Redundancy (TMR), the algorithm computes every block of C two times and perform a comparison. The code to compute a block has been put in a device function that is called by the kernel multiple times. The function returns the value of the element of the block corresponding to the thread. The kernel stored the two values in a local variable. As a result, the additional required memory consists on the remaining part of the stack of the running threads. After double computing of an element, every thread will compare them. If the result is different, a shared variable is automatically set which is used to signal an error to the whole block. Then, the threads need to synchronize in order to read the signaling variable. in such a case that the variable is set because an error has been detected, the value of the element for each thread is computed a

---

```

1: procedure MATMUL_DWC(matrix A(dev), matrix B(dev), matrix C(dev))
2:   Allocate memory for matrix D(dev)
3:   Allocate memory for integer devErrorsCnt
4:   blockSize ← dim3(b,d)
5:   gridSize ← dim3((p+b-1)/b, (m+b-1)/b)
6:   errorsCnt ← 1
7:   while errorsCnt > 0 do:
8:     MATMULKERNEL <<< gridSize, blockSize >>> (A(dev), B(dev), C(dev))
9:     MATMULKERNEL <<< gridSize, blockSize >>> (A(dev), B(dev), D(dev))
10:    *devErrorsCnt ← 0
11:    CUDADEVICESSYNCHRONIZE()
12:    COMPAREKERNEL <<< gridSize, blockSize >>> (C(dev), D(dev), devErrorsCnt)
13:    errorsCnt ← *devErrorsCnt
14:    CUDADEVICESSYNCHRONIZE()
15:  end while
16:  Free memory of D(dev)
17:  Free memory of devErrorsCnt
18: end procedure

```

---

Fig. 3.46 The algorithm of matrix multiplication with DWC.

third time and the most frequent result is stored in the result buffer. Therefore, we do not need to synchronize the thread anymore. The comparison of the voting system is done per element. As a result, every thread can perform it autonomously. Figure 3.47 is representing the pseudo code of the function computing the single element of a block.

At the end, the algorithm represented in Figure 3.44 has been extended in to a larger buffer order to acquire the ABFT implementation of matrix multiplication.

We implemented a solution by maintaining the additional rows and columns used as check-sums in separate buffers and read from or write to these buffers when needed. In such a way, only  $2*m+2*p$  additional floats registers are needed and no copy operation is required, adopting the buffer scheme as in it presented in Figure 3.48. To elaborate more, In order to compute row of A or the check-sum column of B, we can do a sequential reduction optimizing the time complexity, since the computation is performed in very separated thread.

### Fast Fourier transform: plain and fault tolerance algorithm

The implementation of the FFT relies on the Cooley-Tukey algorithm and it can be formulated by the butterfly network in which every two samples of iteration are given by the two samples of the previous iteration in the same position. A buffer in

---

```

1: procedure MATMULKERNERL_TMR(matrix A, B, C)
2:   if tidx = 0 & tidy = 0 then:
3:     errorDetected ← 0
4:   end if
5:   valC1 ← ComputeELEMENT(A,B)
6:   valC2 ← ComputeELEMENT(A,B)
7:   if valC1 ≠ valC2 then:
8:     ATOMICADD(&errorDetected, 1)
9:   end if
10:  ---SYNCTHREADS()
11:  if errorDetected > 0 then:
12:    valC3 ← ComputeELEMENT(A,B)
13:    if valC1 = valC2 || valC1 = valC3 then:
14:      valC ← valC1
15:    else if valC2 = valC3 then:
16:      valC ← valC2
17:    else:
18:      valC ← valC1
19:    end if
20:  else:
21:    valC ← valC1
22:  end if
23:  ia ← bidy.b + tidy
24:  jb ← bidx.b + tidx
25:  if ia < m & jb < p then:
26:    C[ia, jb] ← valC
27:  end if
28: end if
29: end procedure

```

---

Fig. 3.47 The algorithm of Matrix Multiplication with TMR-kernel.

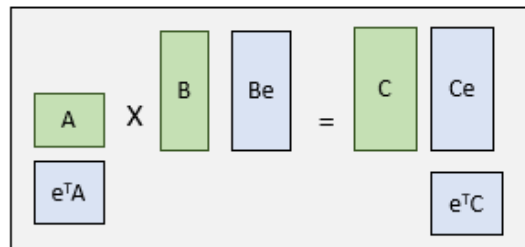


Fig. 3.48 Additional buffers scheme needed for the check-sums for the matrix product ABFT method.

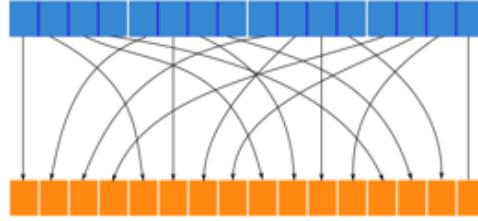


Fig. 3.49 Scheme of the shuffle operation on the FF network to the auxiliary network done in the global memory buffers

---

```

1: procedure FFT(vector x)
2:   Define dimension B of shared memory per block
3:   Sort samples of x according to the index bit inversion rule
4:   for l=0...  $\lceil (\log_2 N) / \log_2 B \rceil$  do:
5:     Launch propagation kernel on x
6:   end for
7: end procedure

```

---

Fig. 3.50 FFT host algorithm executed on the share memory

the global memory is required. Therefore, every thread can read a couple of samples and generate the new couple of samples that replace them.

In used method is based on arranging the samples in blocks properly and operating them separately. There are  $N$  samples in the buffer at each computational step. Therefore, the read and write operations in the global memory are  $N \cdot \log_2 N$ . In a case that the input signal size is more than the shared memory allocated for each block, the usage of the shared memory faces problems. In fact, if a write operation is executed on different portions of the signal storage in the shared memory of separated blocks, at some point a thread will need to access the sample stored in the shared memory to two different blocks. In order to overcome this problem, we shuffled the samples in the global memory every  $N \cdot \log_2 B$  step operations performed in the shared memory. The shuffle operation is not symmetric. Thus, the operation cannot be performed in a parallel way but can be performed during the load of the samples from the buffer of the global memory to the shares memory, realizing the scheme presented in Figure 3.49. Figure 3.50 and 3.51 are representing the pseudo-code of the algorithm developed for FFT.

The performance of the FT algorithm has been executed on a GeForce GT750M having 384 CUDA cores running at frequency of 1058MHz and with a memory transfer rate of 2500MHz. We set the size of the shared memory per block to 4096

---

```

1: procedure FFT_Propagation_Kernel(vector x, iteration l)
2:   Load samples from x in the shared memory
3:   memory doing the appropriate shuffle.
4:   for l=0...  $\lceil (\log_2 N) / \log_2 B \rceil$  do:
5:     Propagate data
6:     Synchronize the threads within the block
7:   end for
8:   Write samples from shard memory to x
9: end procedure

```

---

Fig. 3.51 FFT propagation kernel algorithm executed on the shared memory

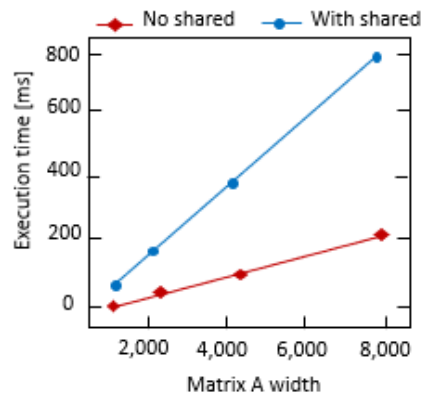


Fig. 3.52 The comparison of the FFT implementation with and without using the shared memory.

complex numbers. While the size of the input data is ranging from  $1024 \times 1024$  to  $8192 \times 1024$  by doubling the first dimension once at a time. Figure 3.52 is representing the acquired results. In both of the implementations, the increase of the time linearly with the size of the input in noticeable. It should be mentioned that the version using the shared memory is always faster and the slope is less comparing to the version without applying shared memory.

In order to obtain the right results, traditional FFT fault tolerance implementation involves full recomputation. In a case of an FFT elaboration on a long signal, if error happens, the time penalty will be extremely long. In order to overcome this issue, we applied a checkpoint scheme in a way that every  $n$  steps, we check the intermediate results for errors. In a case of no error, the intermediate results are stored in the global memory and compute the next  $n$  steps. Differently, we restart the computation of the last  $n$  steps from the last checkpoint. Considering the fact that ABFT based algorithm is a general technique to detect and correct errors, we have to encode in



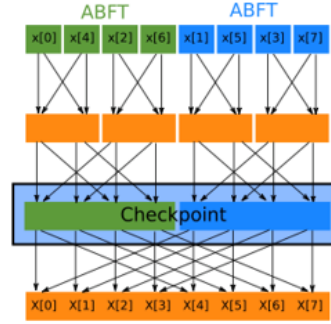


Fig. 3.53 The check-pointing scheme adopted integrating the ABFT algorithm.

---

```

1: procedure FFT(vector x, real  $\epsilon$ )
2:   for i =  $\lceil \log_2 N / \log_2 B \rceil - 1 \dots 0$  do:
3:     mean[i]  $\leftarrow$  mean value of the first sample of each sub-signal
4:     Encode for the checkpoint I from x into y
5:     Partition sub-signals of y in smaller sub-signals in x
6:   end for
7:   Copy x in y
8:   for l =  $0 \dots \lceil \log_2 N / \log_2 B \rceil - 1$  do:
9:     Launch propagation kernel on y
10:    Decode samples of the y vector
11:    m  $\leftarrow$  mean value of y
12:    if m < mean[l] +  $\epsilon$  & m > mean[l] -  $\epsilon$  then:
13:      if l <  $\lceil \log_2 N / \log_2 B \rceil - 1$  then:
14:        Copy y in c
15:      end if
16:    else
17:      Copy x in y
18:      l  $\leftarrow$  l - 1
19:    end if
20:  end for
21:  Copy y in x
22: end procedure

```

---

Fig. 3.54 The FFT algorithm with the ABFT mean based check pointing.

advance all the sub-signals in a bottom-up order. TO elaborate more, firstly, encoding of the signal corresponding to the last decoding is performed. Secondly, encoding the sub-signals corresponding to the second signal subset values is performed. Every time that the encoding is done, we need to shuffle the sub-signals into a smaller group and encode each time we reach a size of the sub-signal corresponding to the size of the checkpoint lever. The scheme is shown in Figure 3.53 while Figure 3.54 is reporting the pseudo code of the developed algorithm.

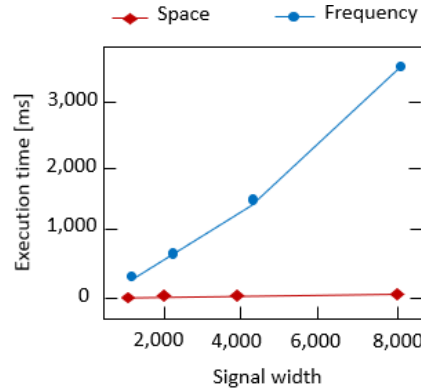


Fig. 3.55 Comparison of the execution times of the Sobel operator implemented in Frequency (blue line) and in space (red line).

### Sobel operator: plain and fault tolerance algorithm

Sobel Operator is a filter that is mainly adopted in image processing and computer vision. The filter allows to compute different transfer functions  $H_x$  from  $G_x$  and  $H_y$  and  $G_y$ , then transforming the signal back and combined according to the main transfer operator. In a mathematical point of view, the Sobel operator is applied to two kernel matrices with the size of  $3 \times 3$  known as convolution matrices of the original source image matrix. Two methods are considered for evaluating the performance of the algorithm: frequency and space. We plot the results in Figure 3.55 with respect to average of 10 executions with a size of signals of  $n \times 1024 \times 1024$  while  $n$  has a range from 1 to 8. The implementation with the direct computational gradient is used for the fault injection analysis since it is much faster with respect to the version with the transformation.

The Sobel operator implemented directly in the space domain, just consists in approximating the gradient in each point by differentiating. It is expected that a single soft error in the logic of the ALU is not affecting all the samples but just one sample of the output because the computation of each sample is independent from the other samples of the transformed signal. By using a DWC scheme, we can identify the corrupted sample and recompute it. If the error occurs in memory, multiple samples could be affected. In particular, all the samples of the output close to the corrupted sample in the input. In such a case, the most convenient way is to recompute the whole output. The DWC implementation of the fault tolerance of the Sobel operator is straightforward and the remaining non linear operation can

be protected from soft errors using a TMR approach. From the implementation standpoint, this algorithm needs the computation of the Sobel operator to be divided into three distinct operations each one with its independent fault tolerant computation.

### 3.4.3 Experimental results

The developed fault injection technique is applied to NVISIA G80 GPGPU model architecture [52] while using the hardware description of the FlexGrip GPGPU [53]. This hardware description is directly implementing the Hardware Description Language of NVIDIA G80 GPGPU. The proposed fault injection method has been applied to Streaming Multiprocessor(SM) architecture which includes a five-stage pipelines architecture consisting Fetch, Decode, Read, Execute and Write stages, supporting the execution of 27 CUDA instructions.

We used Microsemi ProASIC gate library [54]for synthesis of the SM model. We performed two kinds of analysis. Firstly, we evaluate the sensitivity of a single streaming processor regarding transient error. Secondly, we execute various fault injection campaigns on the benchmark applications including Matrix multiplications, FFT and Sobel filter. For both the standard plain algorithm and applying mitigation strategies such as DWC, TMR and ABFT.

#### Streaming processor transient error injection

By synthesizing the SM architecture using Microsemi ProASIC, a netlist of more than 50K gates organized in about 1.5 M logical paths is generated. In order to perform a realistic evaluation of transient errors, we choose a single streaming processor, which includes around 4K gates and registers organized in 238K logical paths and mapped on ProASIC3 Flash-based FPGAs [55]. For evaluating the sensitivity regarding SET, we considered eight different types of SET pulses ranging from 100 ps to 1 ns. We inject randomly 1000 errors for each type of pulses. The results has been classified in four groups: filtered, partially filtered, equal and broadened SETs. Figure 3.57 is reporting the obtained results. It can be noticed that most of the SET pulses with duration less than 0.45ns are filtered. Moreover, by increasing the width of the pulse, the number of unfiltered and broadened pulses increase. However, it is possible to observe that for the SETs with width more than 0.7ns, all the pulses are broadened.

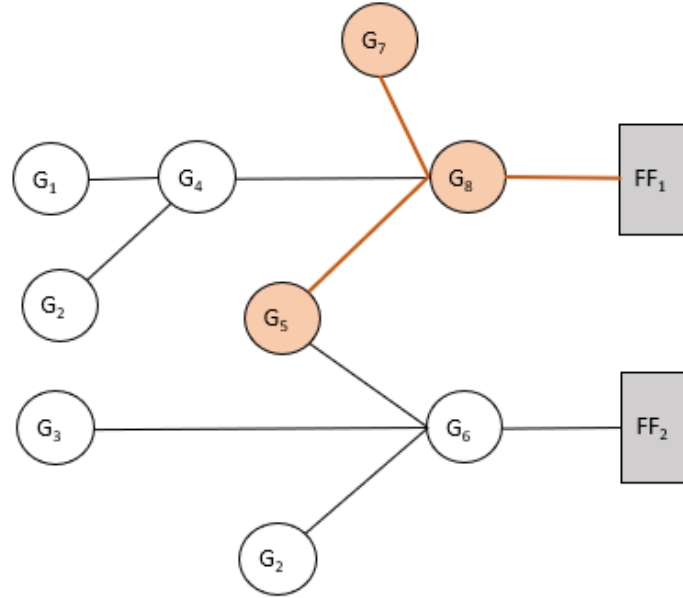


Fig. 3.56 An example of logical cone outputs.

Therefore, all of them are reaching to their respective logical cone outputs. Figure 3.56 represent an example of logical cone considered for evaluation of SET pulses.

Moreover, we investigate the number of computational registers reached by the injection of each SET pulse. It has been observed that the SET pulses are facing a broadening of their width ranging from 5% to 10% of their original width. The transient error profiles acquired by the error injection have been stored into the GPGPU affected register list.

### Application fault injection results

Different applications have been chosen for performing fault injection campaigns such as: a Matrix multiplication between two 16\*16 matrices of integer data, a Fast Fourier Transform(FFT) of a 16\*16 matrix and Sobel filter on a 16\*16 input matrix. We considered four different conditions for implementing the applications: not mitigated, applying Duplication With Comparison(DWC) technique, Triple Modular Redundancy(TMR) and applying Algorithm Based Fault Tolerance(ABFT) mitigation techniques.

The obtained results are presented in Table 3.9, 3.10, 3.11. The percentage of effects are classified as: *application error* which represented the case where the

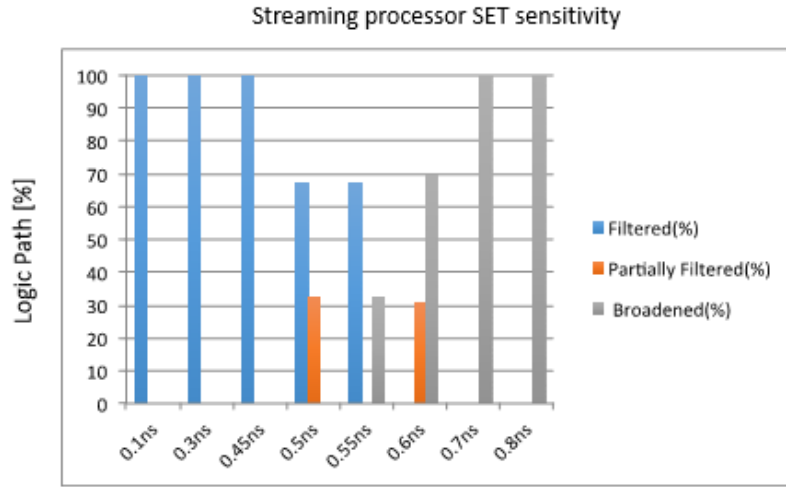


Fig. 3.57 Single streaming processor SET sensitivity overview for injecting 1000 SET pulses.

application generates erroneous results; *Time out* for the case that the application never reaches the generation of the data output; *Silent* which is dedicated to the case that the injected errors do not generate any erroneous results within the executed application and the computed results correspond to normal execution without faults. The average fault injection performance, considering the tested application, may range from 8 to 14 transient errors per minute, while the results are classified as pure combination comparison.

The fault injection speed of the proposed platform has a gain in time that may vary from 2 to 3 orders of magnitude with respect to physical gate level simulation. Moreover, with the proposed method, it is possible to simulate the entire core executing the while application.

As a results, two different scenarios have been observed. Firstly, the application error rate increases with respect to the SET pulse width. Secondly, the increasing ration is not linear since a wider transient pulse is not filtered on more logical paths than shorter ones. Therefore, the application error rate has been increased exponentially.

Moreover, with respect the applied mitigation techniques, the reduction of the fault tolerance capability with respect to the injected transient errors is noticeable. In particular, considering matrix multiplication, the ABFT approach should be able to mitigate all the injected faults. The results shows a different behavior since a transient error may be propagated to many FFs in the computational registers. Therefore,

nullifying the single fault scenario which can be confirmed with the radiation test data obtained previously in [56].

Table 3.9 Matrix multiplication application results fault injection results

SET Pulse[ns]	Application Error[%]				Time Out[%]				silent[%]			
	Plain	DWC	TMR	ABFT	Plain	DWC	TMR	ABFT	Plain	DWC	TMR	ABFT
0.10	0	0	0	0	0	0	0	0	100	100	100	100
0.30	0	0	0	0	0	0	0	0	100	100	100	100
0.45	0	0	0	0	0	0	0	0	100	100	100	100
0.50	8.5	1.7	1.3	1.3	4.2	4.1	4.1	2.9	87.3	94.2	94.6	95.8
0.55	10.6	1.8	1.5	1.5	4.3	4.2	4.2	3.0	85.1	94.0	94.3	95.5
0.60	12.8	1.9	1.6	1.6	4.3	4.2	4.2	3.0	82.9	93.9	94.2	95.4
0.70	14.0	2.4	1.7	1.7	4.3	4.2	4.4	3.1	81.7	93.2	93.9	95.2
0.80	15.3	2.6	1.8	1.8	4.4	4.4	4.4	3.1	80.4	93.0	93.8	95.1
1.00	16.9	2.7	1.9	1.8	4.5	4.5	4.5	3.2	78.6	92.8	93.6	95.0

Table 3.10 Fast Fourier transform application fault injection results.

SET Pulse[ns]	Application Error[%]				Time Out[%]				silent[%]			
	Plain	DWC	TMR	ABFT	Plain	DWC	TMR	ABFT	Plain	DWC	TMR	ABFT
0.10	0	0	0	0	0	0	0	0	100	100	100	100
0.30	0	0	0	0	0	0	0	0	100	100	100	100
0.45	0	0	0	0	0	0	0	0	100	100	100	100
0.50	20.1	12.4	12.3	1.8	0.1	0.1	0.1	0.1	79.8	87.5	87.6	98.1
0.55	20.3	12.6	12.1	2.1	0.1	0.1	0.1	0.1	79.6	87.3	87.8	97.9
0.60	24.5	13.3	13.2	2.9	0.1	0.1	0.1	0.1	75.4	86.7	86.8	97.0
0.70	27.4	15.2	15.0	3.6	0.2	0.2	0.2	0.2	72.4	84.6	84.8	96.3
0.80	31.4	16.8	16.3	4.4	0.2	0.2	0.2	0.2	68.4	83	83.5	95.4
1.00	36.4	17.2	16.9	5.6	0.3	0.2	0.2	0.2	63.3	82.6	82.9	94.2



Table 3.11 Fast Fourier transform application fault injection results.

SET Pulse[ns]	Application Error[%]				Time Out[%]				silent[%]			
	Plain	DWC	TMR	ABFT	Plain	DWC	TMR	ABFT	Plain	DWC	TMR	ABFT
0.10	0	0	0	0	0	0	0	0	100	100	100	100
0.30	0	0	0	0	0	0	0	0	100	100	100	100
0.45	0	0	0	0	0	0	0	0	100	100	100	100
0.50	9.1	0	0	2.1	0.1	0.1	0.3	0	90.8	99.9	99.7	97.9
0.55	10.4	0	0	3.4	0.1	0.1	0.3	0	89.5	99.9	99.7	96.6
0.60	14.5	0	0	5.6	0.1	0.1	0.3	0	85.4	99.9	99.7	94.4
0.70	24.4	0.2	0.1	8.4	0.1	0.1	0.3	0	75.5	99.7	99.6	91.6
0.80	25.4	0.3	0.1	12.5	0.1	0.1	0.3	0	74.5	99.6	99.6	87.5
1.00	25.8	0.3	0.1	14.5	0.1	0.1	0.3	0	74.1	99.6	99.6	85.5

### 3.5 Convergence Single Event Transient Analyzer - CSETA

When a highly charged particle strikes the silicon junction of the device, the transfer of the energy cause a voltage glitch or voltage pulse known as Single Event Transient (SET). The induced SET can propagate through several paths, leading to several SET pulses which may cause multiple upsets in the circuit if they reach to them. When the SET pulse propagate through a logic gate, it may undergo pulse width modulation known as PIPB effect which is due to delay unbalance at different circuit nodes. However, it can be more critical if during the life of SET, the pulse faces a divergence node and propagate through two or more divergence paths which merge at the same convergence node.

#### 3.5.1 SET pulse behavior at convergence point

When the SET pulse is generated inside the implemented circuits, the pulse is propagating through logic nodes and routing interconnections of the implemented circuit. Moreover, during this propagation, the pulse is affected by PIPB. During this propagation, if the pulse encounters a divergence node of the circuit, it will multiply and propagate through several paths. If the multiplied pulses reach to the storage element and be sampled by them, the single generated pulse can create multiple SEUs. In particular, this condition can be more critical if the various propagated SET pulses converge together at the convergence node.

To elaborate more, during the life-cycle, if the pulse traverse a divergence node, it multiplies and propagate through several divergence paths. If the propagated multiplied SET pulses join at the convergence point, the outcome of SET at the convergence brings out a new phenomenon defined as Convergence SET (C-SET). Figure 3.58 shows a simple case where convergence SET can be generated.

The output of SET at the convergence point depends on the PIPB value and the difference between the delay of the two divergence paths. Considering  $SET_A$  as the SET source pulse which propagate through the logics and routing resources of path(A), while broadened or filtered due to PIPB affect as well as  $SET_B$  as the SET source pulse propagated through path(B), then C-SET in the convergence point can be sorted in three groups:

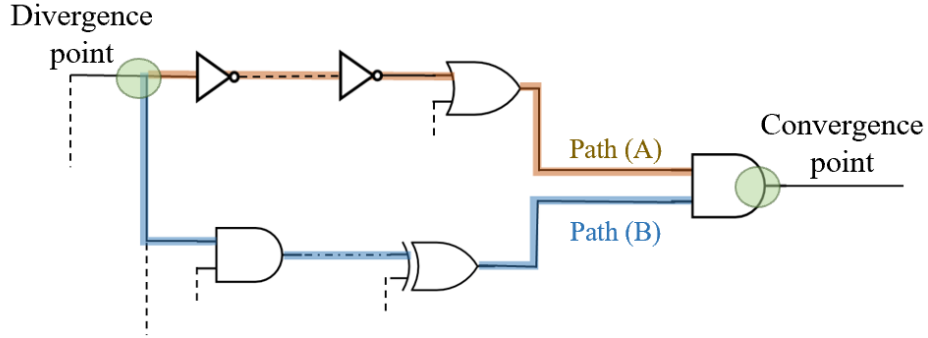


Fig. 3.58 An example of SET encountering the divergence point and convergence point.

1. In a case that the difference between the propagation delay of the two paths is more than the width of the first pulse reaching to the convergence point, the output SET at the convergence point will be provided as two separate pulses with the width equal to  $SET_A$  and  $SET_B$ .
2. In a case that the difference between the delay of the two paths is less than the width of the first pulse reaching to the convergence point, the two pulses will overlap and create a pulse with extremely large width, larger than  $SET_A$  and  $SET_B$ . The most critical condition will be observed when the delay between the two divergence paths is equal to the width of the first SET pulse reaching to the convergence point. In this sub-case, the C-SET observed at the convergence point has a width equal to the sum of propagation delays difference and the width of the second pulse reaching to the convergence point.
3. This third case is dedicated to the condition that the difference between the two paths is minimal enough to create the total overlapping of the two pulses. As a result, the outcome of SET is equal to the maximum width between  $SET_A$  and  $SET_B$ .

Expecting that  $SET_A$  is reaching to the convergence point before  $SET_B$ , Figure 3.59 represents the correlation between the two propagated paths, the different of the delay of the two paths and the condition of the output SET at the convergence point while equation 3.8 is representing the condition of each group. Moreover, Figure 3.60 represents the correlation between the maximum width of SET observed at the convergence point and difference of the delay between two paths. As it has been mentioned, case 2 is representing the worse conditions due to the widening of the SET pulse at the convergence point.

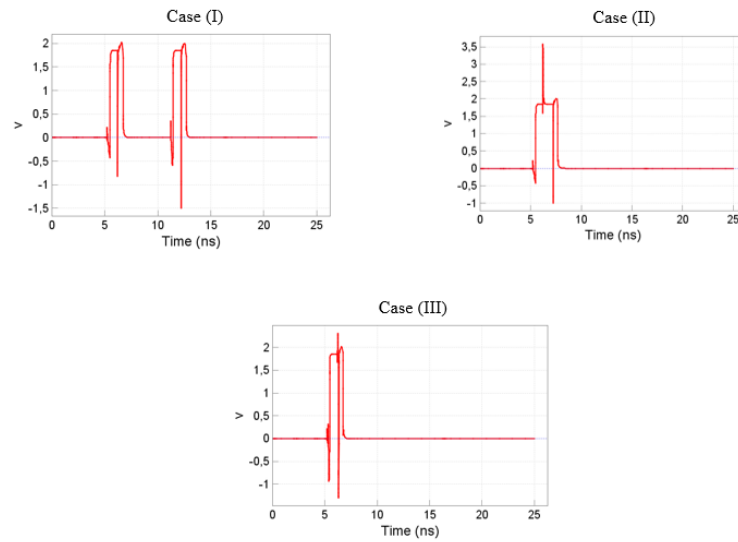


Fig. 3.59 Outcome of SET at the convergence point- C-SET.

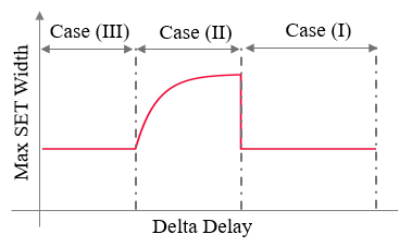


Fig. 3.60 Correlation between maximum width of C-SET and Difference of delays between two paths.

*Case(I) :*

$$Delay_{path(A)} - Delay_{path(B)} > SET_A$$

*Case(II) :*

$$Delay_{path(A)} * Delay_{path(B)} \leq SET_A$$

$$(Delay_{path(A)} - Delay_{path(B)}) + SET_B > SET_A \quad (3.8)$$

*MostCriticalCase :*

$$Delay_{path(A)} - Delay_{path(B)} = SET_A$$

*Case(III) :*

$$Delay_{path(A)} - Delay_{path(B)} \leq SET_A$$

$$(Delay_{path(A)} - Delay_{path(B)}) + SET_B < SET_A$$

### 3.5.2 Integration of CSETA with commercial tools

Since Convergence SET can introduce a more critical situation for the system with respect to SET pulses, it is mandatory to evaluate the behavior of the implemented circuit with respect to Convergence SET. Therefore, we developed an environment for identification of SET propagation considering its convergence condition within a circuit. Therefore, a software tool has developed which is integrated with the modern version of commercial tool for designing Integrated Circuits. Figure 3.61 represents the developed workflow.

The developed Integrated Design Flow (IDF) is linked with the classical design tool chain starting with the Hardware Description of the design, going through synthesizer, mapping and place and route. The IDF starts elaborating the post-layout design and generating Physical Design Constraints (PDC) and Simulation Delay File (SDF) from post-layout netlist. The PDC file contains all the locations of the logic resources and input/output pins of the target FPGA device while SDF contains the delay information of the implemented design. Based on the PDC and SDF files of the design under the test, we developed a tool to extract the Physical Design Description Annotated (PDDA) of the design which contains the delay information of the routing and logics.

In order to analyze the implemented circuit regarding C-SET, we started from the PDDA file, elaborating the physical description of the circuit, extracting all the paths of the design. This extraction has been considered starting from each nodes of

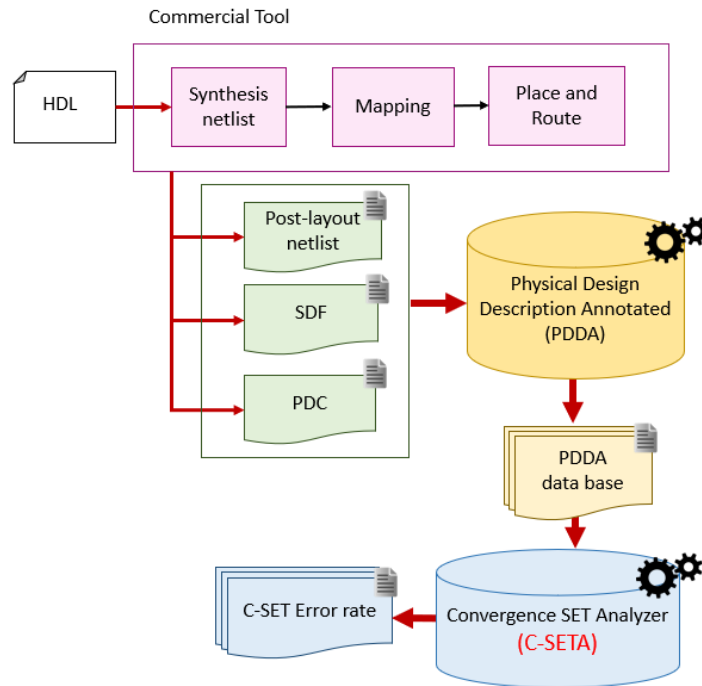


Fig. 3.61 Scheme of developed flow for accurate analysis of C-SET.

the paths until it reaches a storage element or FF. These extracted paths have been elaborated in three phases, as represented in pseudo code of Figure 3.62.

For the first phase, the behavior of SET propagation through the path has been evaluated. To do so, the propagation behavior of each logic cell and routing resources connecting the logics together are studied. As a result, we calculated the Pulse Induced Propagation Broadening (PIPB) effect for all the extracted paths. As a next step, we elaborate the extracted path in order to identify the possible condition of occurrence of C-SET. Therefore, we develop a tool that first of all classifies all the logic cells of the design which has a divergence characteristic. To elaborate more, the tool extracts all the logic cells having more than one outputs. A logic cell with more than one outputs generates several branches in the main path. The tool continues to extract all the branches connecting to the divergence point, and filtering the branches which are reaching to the common logic cells, defined as Convergence point. As a result of this phase, we extracted all the paths starting from the common divergence point and converging in a common logic node which leads to all the possible locations of occurrence of C-SET. However, reaching to the same convergence node is not enough to generate C-SET phenomenon. Therefore, the

```

//Initialization

Foreach Design- Path:
    Foreach Logic-Cell in the path:
        if Logic-Cell is a divergence node:
            DIV-Branches = extract-Branch(DIV-node);
            CONV-Branches = extract-C-SET-Branch(DIV-Branches);
            C-SET-Cases= Timing-Analysis (CONV-Branches);
// First Function
Function extract-Branch(DIV-node)
    Output-of-DIV-node = DIV-node;
    extract-Branch (DIV-node);
    Return DIV-Branches;
// Second Function
Function extract-C-SET-Branch (DIV-Branches)
    For i=0; i< DIV-Branches-number
        For j=i; j< DIV-Branches-number
            if ( DIV-Branches[i].Last-node==DIV-Branches[j+1].Last-node)
                CONV-Branches [CONV-Branch-No][0]=DIV-Branches[i];
                CONV-Branches [CONV-Branch-No][1]=DIV-Branches[j+1];
                CONV-Branch-No= CONV-Branch-No+1;
    Return CONV-Branches;
// Third Function
Function Timing-Analysis (CONV-Branches)
    For i=0; i< CONV-Branches-Number; i++
        delta-Delay = Delay-CONV-Branch[i][0]- Delay-CONV-Branch[i][1];
        if (delta-Delay < SET-Pulse);
            Report Overlapped SET;
        Else if (delta-Delay < SET-Pulse)
            Report Double-SET;
    Return C-SET-cases;

```

Fig. 3.62 The pseudo code for the identification of the SET width and amplitude in a post layout circuit.

next phase is dedicated to analyze the extracted branches regarding timing to analyze the extracted branches regarding timing to recognize all the authorize location of occurrence of C-SET.

As it has been mentioned before, the PDDA file of the implemented design contains timing information. Therefore, the tool is linking between the extracted branches and the timing constraints of PDDA file and classifying the branched based on timing. Starting from the divergence point until the defined convergence point, for each single branch, the tool is calculating the delay of the path, which takes into account the delay of the logic node and also the routing connecting the logics. At the end of this phase, we extracted all the branches between the divergence and convergence point and the delay of each branches. Based on the calculated delay, we classified the branches. As it is shown in Figure 3.58, if we consider two branched of A and B which start from the divergence point and end at the convergence point, the tool is calculating the delay regarding path(A) and path(B) and the different between them. Depending on the different of delay, the C-SET at the convergence point is estimated which can be classified in several groups. If the delta delay of paths are less than the duration of the expected SET, single SET will be observed at the convergence point. However, due to the overlapping of SET pulses, the duration of observed C-SET is wider as much as the different of delay between two paths. While, if the delta delay of two paths are more than the duration of the expected SET, the SET propagation through two paths will cross the convergence point separately which generated double SET at convergence point which has the distance equal to delay delay of the branches shown in equation 3.8.

### 3.5.3 C-SETA on Flash-based FPGAs

The proposed design flow has been experimentally evaluated by means of SET static analysis using A3P250 Flash-based FPGA manufactured by Microsemi having 6,144 logic Versatiles. We used Libero soC commercial design flow to generate the PDC, SDF and post-layout netlist. We select six circuits with various complexity from ITC99 benchmark collection [22], a Cordic core and RISC microprocessor. The characteristics of the selected circuit such as the number of Versatile used as Logic Function or Flip-Flop and maximal working frequency have been reported in Table 3.12.



Table 3.12 Characteristics of the original benchmark circuits

Circuits	Versatile[#]	FFs[#]	Frequency[MHz]
B05	415	66	47
B09	493	67	46
B12	565	123	48
B13	162	50	52
CORDIC	956	240	45
RISC	1,401	1,156	42

We analyzed the circuits SET sensitivity using the developed tool able to evaluate the convergence SET. For the purpose of our experiments, we performed the analysis considering three type of SETs (0.3, 0.6 and 0.8 ns), 5000 SET injection for each type of SET pulses. Please consider that SET width lower than 1 ns corresponds to the most probable event generated by heavy ion strike on Flash-based FPGA 130 nm technology [57].

As a first step, we performed the analysis of sensitivity of the implemented circuit regarding normal SET using SETA, explained in the previous section. In Table 3.13, the results of SET analysis have been reported in terms of status of each Flip-Flop as:

1. Filtered: it stands for number of Flip-Flops where SET pulses are filtered before reaching to the Flip-Flops.
2. Partially Filtered: representing the case that PIPB effect is between 1 and 0 which causes partially filtered of SET pulse.
3. Broadened: which reports the number of Flip-Flops in which SET pulses are reaching to them by PIPB value more than 1, which introduced the broadening case of SET pulses.

At the end, we reported the number of possible cases for C-SET in the implemented circuits. It is possible to observe that the number of possible C-SET events is directly proportional to the arithmetic circuitry used by the netlist.

We continued the analysis by applying our proposed flow to evaluate in details the sensitivity of the circuits under the test regarding C-SET. As a result, we classified the possible C-SET events in three groups:

Table 3.13 Comprehensive SET sensitivity using static analysis tool- 5000 SET pulses lower than 1 ns are injected.

Circuits	Filtered[#]	Partially Filtered[#]	Broadened[#]	C-SET[#]
B05	9	3	8	9
B09	3	6	11	18
B12	1	7	13	8
B13	14	8	7	38
CORDIC	12	28	39	42
RISC	204	184	196	56

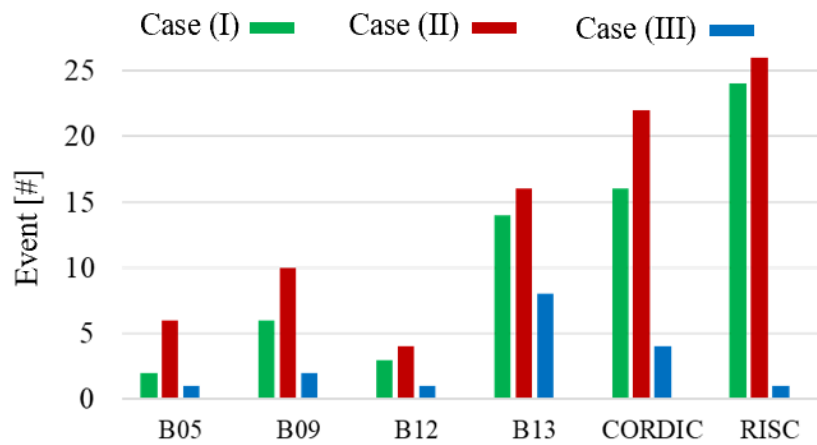


Fig. 3.63 Classification of C-SET in terms of criticality.

- Case 1: representing the number of cases which two separate SET pulses are expected.
- Case 2: as the condition of overlapped SET which one single SET pulse with drastic increasing of pulse width is expected.
- Case 3: representing the overlapped SET while the width of C-SET is equal to the maximal pulse width of propagated SETs through divergence paths, presented in Figure: 3.63.

Between these three classified cases, Case 1 and 3 are not introducing more critical condition regarding SET sensitivity comparing to normal SET. In fact, these cases can be treated the same as normal SETs in terms of mitigation solutions, such as guard gate approaches introduced in the following chapter. The most critical case

in case 2 which outcome SET at the convergence point has wider width comparing to other cases. Therefore, a mitigation solution with higher filtering capability should be applied to the circuit in order to cover the filtering of normal SETs and C-SETs.

#### **3.5.4 Research advancement on Single Event Transient Analyzer**

In order to provide an accurate analysis of the reliability of the design regarding SET implemented on different modern VLSI technologies such as Flash-based FPGA, SRAM-based FPGA and GPGPU, a new CAD tool has been developed. The developed CAD tool, SETA, is integrated with the standard FPGA design flow. The developed tool takes into account the pulse propagation behavior through the routing and logic resources of the design and evaluate the pulse broadening and filtering effects while propagating through the implemented circuit. The developed tool has been known as the first tool applicable to large industrial circuits in order to provide the sensitivity of the implemented design.

## Chapter 4

# Mitigation of Single Event Transient

The aggressive scaling trend in nanometer technologies has significantly impacted the rates of Single Event Transient(SETs) faults within the electronic circuits. Several mitigation solutions have been proposed in order to make the modern complex system robust against Single Event Transients effect. Among the studies dedicated to FPGA technologies, FPGAs with Flash-based configuration cells are mainly addressed since their configuration memory cells are essentially immune to bit-flips. Therefore, several studies are proposed for mitigation of the occurrence of SETs. As a tradition fault-tolerant strategy, Triple Modular Redundancy(TMR) can be mentioned which is based on the redundancy concept [58]. Some techniques have been proposed based on the replication design methodology by using time or spatial redundancy. Although, techniques based on redundancy introduce delay, power and area overhead to the system [59]. To overcome these disadvantages, recent methods perform the sensitivity analysis of the circuits to identify the sensitive nodes and apply mitigation solution to them, known as selectively mitigating [60] [61] [62].

Mitigation solution based on filtering are applied to many designs without considering timing and resource overhead constraints [63]. Other solutions are based on analytical tools which has been confirmed radiation-test experimental analysis. These tools are providing a possible alternative to time-consuming physical design simulation [64]. ( I would like to complete this part\*\*\*).

In order to mitigate the Single Event Transient affecting Flash-based FPGAs, we proposed two mitigation solutions: The first one is based on the filtering Guard Gate which is applied to selective sensitive points of the circuits based on the performed

SET analysis. However, this method is suffering the disadvantages of introducing the timing and area overhead to the design under the test. Secondary, we proposed a mitigation solution based on the charge sharing concept which overcomes the problem of introduced delay in the circuits since the method is capable of filtering the SET pulses with zero-timing overhead.

## 4.1 Guard Gate mitigation

In order to mitigate the SET pulses in the circuit under the study, we propose a new mapper algorithm in order to selectively introduce SET-filtering scheme focusing on optimizing the circuit performance and reducing the overall SET sensitivity. This techniques is providing an enormous benefit versus already used implementation tools that apply SET-filtering and guard-gate for all the user memory or Flip-Flop resources [60] [65] [66].

The developed Integrated Design Flow (IDF) includes three groups of tools: SET Analyzer (SETA, which has been explained in details in chapter 3), a netlist modifier and a design physical implementation tool. Figure 4.1 is representing the developed IDF.

The developed environment is starting with the classical design tool chain including synthesis, mapping and place and route tools. From the classical tool chain, the IDF elaborates the post-layout design and adopting a technology radiation sensitivity data provided by FPGA producers or generated by preliminary radiation-test characterization, including elementary radiation sensitivity per each type of logic functions or routing segment implemented by FPGA [67].

The SET mitigation flow has three steps:

1. SET Analysis (SETA): The circuit under the study is analyzed regarding Single Event Transient by SETA tool, explained in details in the previous chapter. As a result of SETA tool, two reports are generated. Firstly, a file reporting a list of maximal glitched observed at the input of each Flip-Flop. Secondly, a file reporting the location of Flip-Flop cells and their logic path delays.
2. Netlist Filter Mitigation insertion: The synthesized netlist of the circuit under the study has been modified by inserting filtering circuit to improve the SET

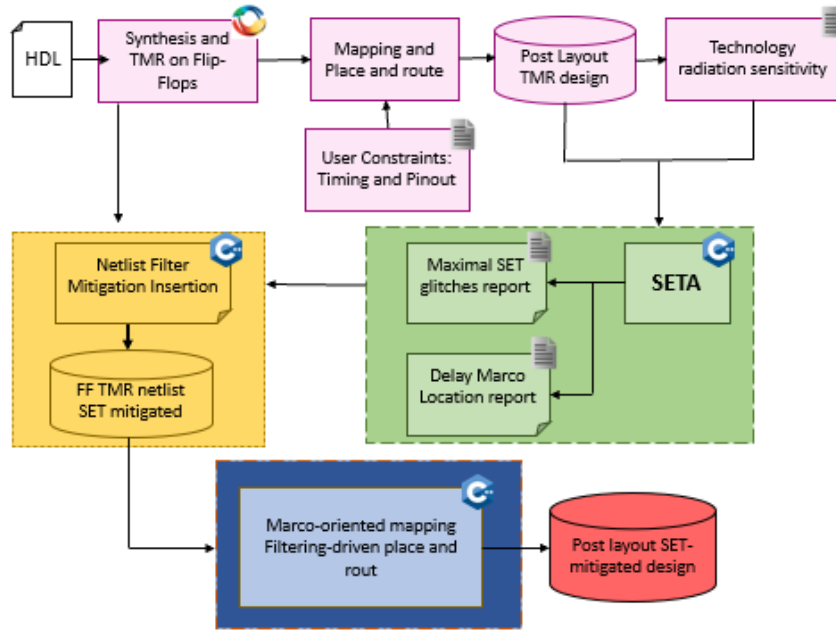


Fig. 4.1 The Overview of the SET-aware mitigation flow including the SET propagation analysis, the Netlist filter mitigation insertion and the marco-oriented mapping and filtering-driven place and route.

mitigation. The insertion has been done considering the post-synthesis Verilog description of the design. To elaborate more, based on the performed SET analysis performed by SETA in the previous step, the path which reports the maximal glitched are modified and the filtering logic are inserted.

- Physical implementation: The modified netlist goes through the mapper and place and route tool. The mapper elaborates the netlist optimizing the SET filtering capability by creating proper design macro. At the end, the place and route algorithm specify the logic versatile locations and the routing segments that optimize the SET filtering of the design without penalize the circuit performances.

#### 4.1.1 SET propagation analysis

As it has been elaborated in chapter 3, SETA tool has been adopted to the design under the test in order to provide the sensitivity report of the design. Therefore, we start with the generation of source SET pulses, propagating the generated pulse

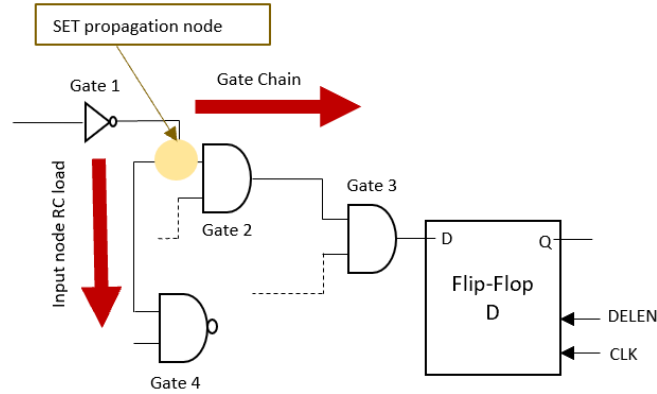


Fig. 4.2 The SET propagation method including the PIPB computation on the propagation node.

through the paths of the design. During this propagation, the Propagation Induced Pulse Broadening (PIPB) model has been applied which elaborated the transformation of the pulse shape along the traverse path up to a Flip-Flop or other storage element for each logic cone. This PIPB coefficient is obtained based on the number and type of logic gates spanning from the SET sensitive nodes to the drain Flip-Flop and logic input node Fan-in. Figure 4.2 represents how the PIPB coefficient of the propagation node is calculated considering the existence of Gate2, Gate3 which Gate4 is representing the Fan-in. The details of this steps has been elaborated in Chapter 3.

#### 4.1.2 Netlist filter mitigation insertion

The typical SET mitigation techniques for Flash-based FPGAs applies Guard Gate (GG) logic gate structure at the input of all the Flip-Flops [62]. Figure 4.3 represents the conceptual scheme of the Guard Gate techniques.

The traditional techniques of inserting Guard Gate filtering has two main disadvantages: Firstly, since the Guard Gate filtering is inserted in the inputs of all the FFs, it has a drastic area overhead. Secondly, inserting Guard Gate logics which is at least 6 gates for each Flip-Flops introduces a performance degradation, Figure 4.3.a.

In order to overcome the mentioned drawbacks of the traditional Guard Gate techniques, we developed an algorithm which insert filtering logic not for all the Flip-Flops but for all the logic gates shared between logic cones, as illustrated in

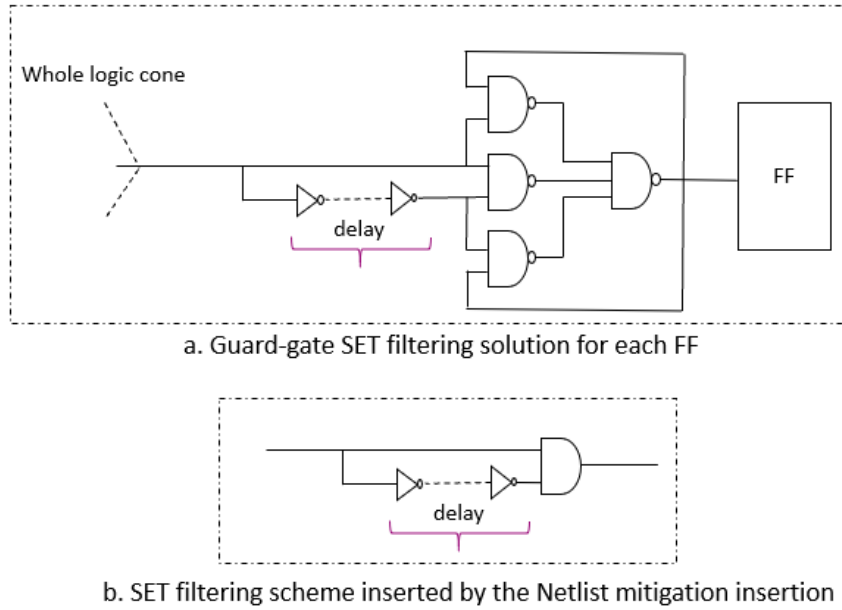


Fig. 4.3 Traditional Flip-Flop based guard-gate and SET-Filtering solution(a)compared to the SET-filtering scheme inserted by the netlist insertion mapper on logic gates shared by logic cones(b).

Figure 4.3.b. The algorithm identifies the points in the netlist for inserting filtering logics and calculates the effective SET filtering delay.

To elaborate more, firstly, the algorithm calculates the broadening coefficient for the used resources with respect the the calculated maximal SET glitches report. The broadening coefficient,  $\Delta BG_i$  for each logic gates where  $i$  is the index of the gate and  $\Delta BG_j$  for each routing net while  $j$  is representing a specific routing segment.

Please notice that a positive value of the coefficient is representing a broadening effect while a negative value is presenting the filtering of SET. The computation provides further information of the broadening contribution of each logic total path, represented by  $\sum B$ . The broadening contribution is calculated for each Flip-Flop element and for all the gates shared between two or more logical paths.

Figure 4.4 illustrates the applied proposed techniques to the portion of the circuit. Considering real cases, the growing maximal SET broadening width contribution  $\Delta b$  and the total SET broadening contribution  $\sum B$  are computed for each logic gate and routing segment in the target design. Without applying any filtering logic, the maximal SET width at the input of the Flip-Flop A is 1.2ns and Flip-Flop B is 1.8ns.





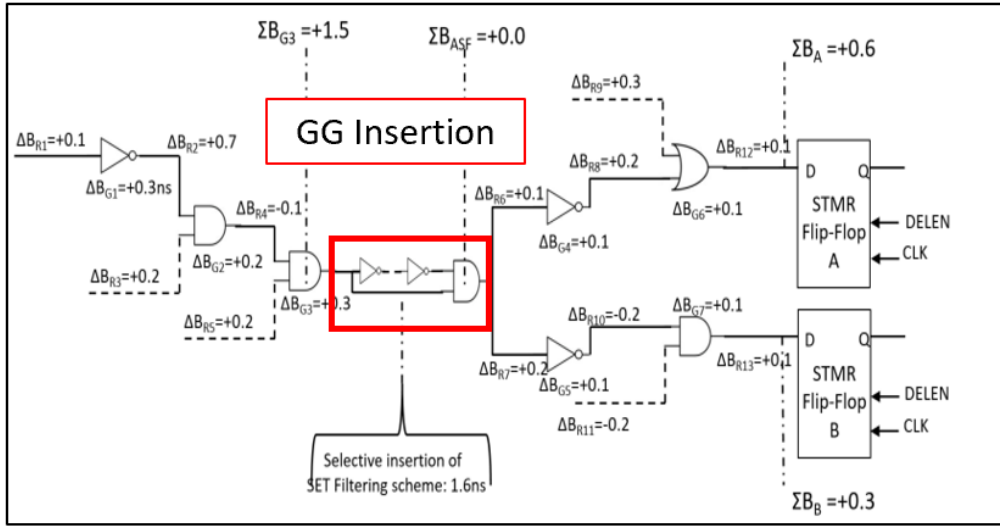


Fig. 4.5 Application of the netlist mitigation algorithm second phase to a circuit portion: the insertion of the filtering scheme on the shared logic path.

for placing in the FPGA area corresponding to the gates used by the netlist insertion algorithm. Therefore, considering  $MB_0, MB_1, \dots, MB_n$  the set of macro-blocks generated during the netlist modification phase, where each  $MB_i$  has an associated dimension (height and width), the goal of the placement algorithm is to provide a placement rectangular area  $R$  for each of the macro-block in a way that:

1. The resource of the area  $A_i$  used by the macro-block has a sufficient resources.
2. Two  $MB_s$  do not overlap.
3. The resources within the  $MB_s$  fulfill the timing delay defined during the netlist insertion phase.

The pseudo code of the developed algorithm is presented in Figure 4.6. The algorithm start with initialization of the design macro-block. The information regarding macro-blocks is provided as a result of SET analysis performed by SETA and the netlist insertion algorithm back annotates it. The algorithm continues by clustering the various  $MB_s$  with respect to the number of Inverters and creating a set of block To Be Placed (TBP). As a last step, the algorithm focuses on the allocation of the clustered  $MB_s$ . To elaborate more, a  $MB$  is added to the Area Cluster (AC) only if it is able to maximize its filtering capability considering also the position of the other placed  $MB_s$ .

---

```

1: initialization of design macro-blocks (MB)
2: clustering (MB)
3: for TBP to total MB
4:     while Dimension(TBP)>1
5:         select (MBi,MBj) for maximizing  $F_k$ 
6:         add MB to Area Cluster(AC)
7:         update AC available regions
8:     end while
9: end for

```

---

Fig. 4.6 The global placement implementation algorithm.

#### 4.1.4 Guard Gate Mitigation on Rad-Hard RTG4 Flash-based FPGAs

In order to fulfill the increasing aerospace requirements in relation to the Total Ionizing Dose(TID), a new radiation-hardened Flash-based FPGA family, RTG4, has been recently manufactures [58]. RTG4 technology is a TID tolerance more than 100 krad due to the complementary or C-Flash and configuration cell [59]. Therefore, this family of Flash-based technology is able to tolerate higher level of Ionizing Dose rather than the previous N-Flash.

Considering the transient radiation effects, RTG4 family is offering an embedded SEU and SET mitigation scheme that rely on Triplicated Flip-Flop architecture and internal SET mitigation. However, the available implementation tool is not allow the designer to apply the proper redundancy and SET filtering setup. Therefore, we choose RTG4 as a modern Radiation-hardened technology for applying our developed environment and selectively SET-filtering scheme. In order to do this, we evaluate the elaborate of the RTG4 cell library and it is interfacing with the available tools.

#### Background on RTG4 Architecture

The RTG4 Flash-based FPGA technology is based on an array of Flash-technology based radiation tolerant logic elements embedding some hard ASIC blocks such as RAM memory modules and DSP blocks. The embedded registers are capable of mitigation Single Event Transient by inserting filtering logic. On the other side,

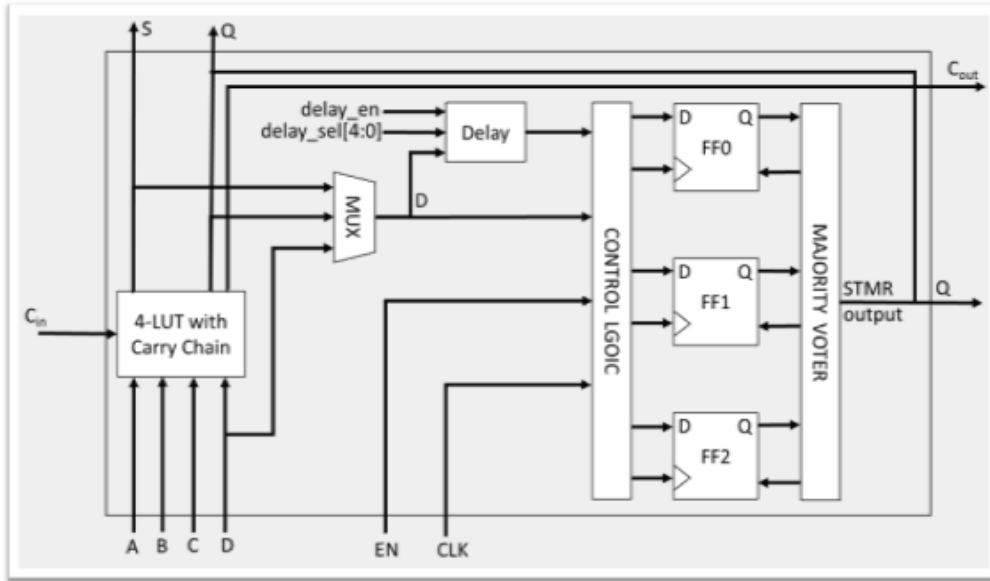


Fig. 4.7 The functional block diagram of logic element of the RTG4 Flash-based FPGA family.

memories have a built-in error detection and correction mechanism(EDAC). The major resources of the RTG4 FPGA architecture are: logic elements, interface logic elements and I/O modules [68]. For applying our developed environment, we focused on the logic elements that compose the larger part of the FPGA resources [69].

Figure 4.7 represent the RTG4 logic element which consists on 4 inputs Look-up Table(LUT), a self-corrected Triple Modular Redundancy(S-TMR) Flip-Flop and a dedicated carry chain.

The 4-LUT with carry chain logic can be configured to any 4-input combination function where the LUT output is XORed with the carry input signal( $C_{in}$ ). When the LUT is implementing a combinational function, output S is the principal output. The carry chain has a specific hardwired interconnection between the logic elements able to reduce the propagation between the logic elements which is reducing the propagation delay through the carry chain. The new advantages and offers of RTG4 technology is the SET mitigated asynchronous and self corrected TMR-D Flip-Flop(S-TMR). Particularly, each STMR Flip-Flop has an Asynchronous majority voter logic that ensures SEU immunity when the SET pulse width at the input D of the functional logic block is comprised within the user defined SET filtering delay coefficient *delay\_sel*. Looking at the implementation tool, when the SET filtering

Table 4.1 Characteristics of the implemented circuits

Circuit	4LUTS[#]	DFF[#]	Area Overhead[%]	Max Clock Period[ns]
B05	205	46	-	10.49
<i>B05_SET</i>	205	46	0	11.22
<i>B05_GG</i>	481	46	134	16.31
<i>B05_SEL_MAP</i>	255	46	24	12.10
B12	378	119	-	8.47
<i>B12_SET</i>	378	119	0	9.40
<i>B12_GG</i>	1,092	119	189	15.45
<i>B12_SEL_MAP</i>	502	119	33	9.82
B14	1,607	216	-	21.14
<i>B14_SET</i>	1,607	216	0	22.09
<i>B14_GG</i>	2,903	216	81	28.44
<i>B14_SEL_MAP</i>	1,895	216	18	22.20

is activated in the tool, the filtering logics are inserted which can lead to the huge reduction in the timing performances. However, the commercial tool is not able to provide the effective width of the SETs for filtering. Therefore, we proposed to apply our developed environment to acquire an effective mitigation of SET on RTG4 family.

## Experimental Results

We applied our developed environment to RTG4 RTG4G150-CG1657 Rad-Hard Flash-based FPGAs and we select three different benchmark from ITC99 benchmark circuit [22]. We implemented the benchmarks in four different versions: original unhardened, commercial tool-based SET filtering(*\_SET*), Flip-Flop-based guard-gate solution(*\_GG*) and the proposed approach(*\_SEL\_MAP*). Table 4.1 is reporting the characteristic of the chosen circuits. Regarding the mitigation, the commercial tool is set for filtering of SET with 0.6 ns while we set the filtering for pulses with the width of 1.5 ns.

As it is reported, the proposed solution minimize the area overhead reaching an average of 25% with respect to the original circuit without mitigation. This percentage is extremely effective if compared with the guard-gate solution, which has an overhead of about 135%.

Table 4.2 SET fault simulation results

Circuit	Observed SET[%]	Filtered SETs[%]
B05	85	15
<i>B05_SET</i>	32	68
<i>B05_GG</i>	12	88
<i>B05_SEL_MAP</i>	1	99
B12	87	13
<i>B12_SET</i>	33	67
<i>B12_GG</i>	16	84
<i>B12_SEL_MAP</i>	3	97
B14	89	11
<i>B14_SET</i>	36	64
<i>B14_GG</i>	15	85
<i>B14_SEL_MAP</i>	3	97

The capability of SET mitigation techniques are reported in term of SET fault simulation. We considered the injection of 10,000 SETs in random locations and sensitive nodes of the benchmark circuits. The width of source SETs are considered between 0.01 ns and 1.00 ns and they have been injected in all the possible sensitive points of the netlists. Table 4.2 is reporting the result of SET sensitivity evaluation. The results are classified as: SET causing erroneous circuit behavior(Observed SET) and Filtered SETs.

As it is illustrated in the results, our proposed environment provides a mitigation techniques which is 4 times better than guard-gate solution. Please notice that the actual results are related to the commercial solution that does not allow SET filtering delay for SETs with width larger than 0.6ns. However, considering the timing characteristic, it is feasible that all of the techniques is introducing timing overhead and performance degradation. Therefore, as a next step, we propose a mitigation solution with zero timing overhead.

## 4.2 SET Mitigation by adding Charge Sharing logics on Flash-based FPGA

Several SET mitigation solutions have been proposed. Some solutions focused on the physical layout modification while the other solutions are focused on the modification

of routing segments without Flip-Flops placement modification. However, these solutions require the reconfiguration of the resources which leads to the modification of logic and routing segment and affecting the overall circuit performances. Solutions based on filtering structure insertion which as it has been mentioned previously, they introduce heavy performance and hardware resource overhead.

To overcome these disadvantages, we proposed a new mitigation solution which is not introducing timing degradation. This mitigation technique is based on charge sharing gate insertion into the circuit netlist which is decreasing the sensitivity of the nodes. This method is known as the **First** approach able to implement SET filtering on Flash-based FPGAs without any timing penalty. The developed mitigation approach is based on two concepts: Firstly, the application of the charge-sharing phenomena on the Flash-based FPGA logic element. Secondly, the control of the routing segment buffering voltage threshold level.

Charge sharing is a concern for CMOS technology. This is principally due to the higher packing densities, reduced nodal charge and space between device resources. Specifically, for nanometer Flash-based FPGAs, the proximity of device nodes results in charge collection in multiple logic switches when a single heavy ion strike a node. The phenomena results in different transient pulse shapes related to the LET absorbed by the switch junction. The proposed mitigation solution modifies a place and routed circuit on Flash-based FPGA by inserting programmed logic gates in ad-hoc netlist nodes. The insertion is performed in order to distribute the charge collection through a circuit logical path, reducing the amplitude and the width of the transient pulse traversing the logic elements. Since the insertion of charge sharing gates increases the fan-out of the selected nodes, we control the insertion while avoiding the increasing of node delay. This is possible due to the lower buffering threshold level provided in FPGA routing node: if the fan-out is below the threshold of the technology under the study, the delay of the traversing signal is not affected. On the other side, the added logical node will reduce the PIPB effect. Therefore, nullifying the SET effect before reaching a sampling node.

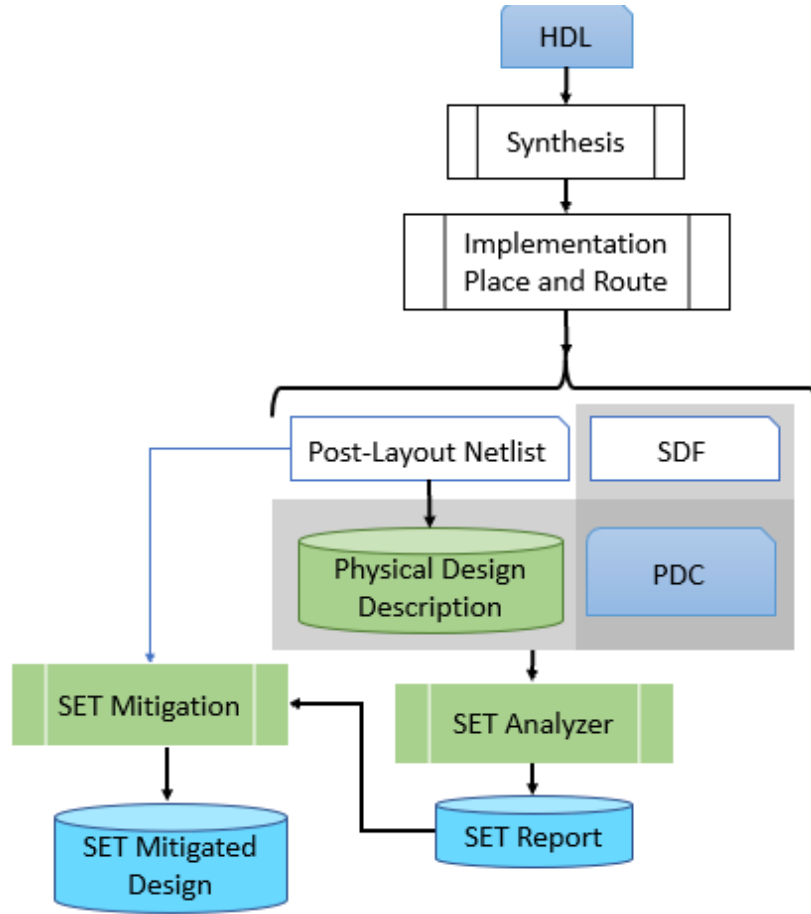


Fig. 4.8 Overview of the developed flow including SET analysis and charge sharing mitigation.

#### 4.2.1 Proposed design flow

To mitigate the SET induced by radiation particles striking silicon structure of Flash-based FPGA devices with charge sharing mitigation technique, the proposed design flow is illustrated in Figure 4.8.

The flow starts with the output files from the commercial/standard FPGA design flow, which includes the Post-layout Netlist, SDF and the Physical Design Constraints(PDC). Firstly, the Post-Layout Netlist is converted to a format designed in-house named Physical Design Description(PDD) file which stores the circuit in a graph representation. As a next step, the timing information extracted from the SDF and the placement information within the PDC, the SET analyzer tool(SETA, which has been explained in details in chapter 3) is executed. As a results, SETA tool



---

```

1:  //Initialization Phase
2:  Netlistorig = Verilog_load();
3:  Netlistsol = {0};
4:   $\forall$  Node  $n \in \text{Netlist}_{\text{orig}} \rightarrow \text{PIPB}[n] = \text{SET\_Report}[n]$ ;
5:   $\forall$  Path  $p \in \{\text{FF\_I}, \text{FF\_o}\} \rightarrow \text{RC}_{\text{node}}(p) = \{0\}$ ;
6:  //1. RC load computation
7:   $\forall n \in P$  do
8:    for  $i \in \text{output\_nets}(n)$ 
9:      if  $i$  is not buffered
10:         $N_{\text{NB}}(i) = \text{Time\_Unbuffered}(n, \text{Netlist}_{\text{orig}})$ ;
11:      else
12:         $N_{\text{BU}}(i) = \text{Time\_Buffered}(n, \text{Netlist}_{\text{orig}})$ ;
13:
14:     $\text{RC}_{\text{node}}(n) = (\sum N_{\text{NB}}(i) + N_{\text{BU}}(i)) / \text{Fan\_out}(n)$ ;
15:  //2. Charge Sharing Computation
16:   $\forall$  Node  $n \in \text{Netlist}_{\text{orig}} \rightarrow \text{CS}[n] = \{0\}$ ;
17:   $V_{\text{path}}[n] = \{0\}$ ;
18:  for  $p \in P : \text{CS}[n] = V_{\text{path}}$ 
19:     $\forall n \in P \rightarrow G(n) = \text{interpolate}(\text{PIPB}[n], \text{RC}_{\text{node}}[n])$ ;
20:     $\text{PIPB}_{\text{min}} = \text{Max}(\text{PIPB}(p))$ ;
21:     $H_{\text{min}} = \text{Max}(\text{RC}_{\text{node}}(p))$ ;
22:    for  $i \in n$  generate binary permutation  $S$ 
23:       $H = \text{card}(G(n))$ ;
24:       $\text{PB} = \text{global\_PIPB}(p, S)$ ;
25:      If  $\text{PB} < \text{original\_PIPB}(p) \ \&\& \ H \leq \text{original\_card}(p)$ 
26:         $H_{\text{min}} = H$ ;  $V_{\text{path}} = S$ ;
27:  //3. Modify Netlist
28:   $\forall$  Node  $n \in \text{Netlist}_{\text{orig}}$ 
29:     $\text{Netlist}_{\text{sol}} = \text{add\_CS\_structure}(V_{\text{path}})$ ;
30:  //4. Export Verilog netlist
31:  Export_Verilog( $\text{Netlist}_{\text{sol}}$ );

```

---

Fig. 4.9 The charge sharing mitigation algorithm for Flash-based FPGAs.

generated a SET report which contains the information regarding SET sensitivity for each Flip-Flop in the design, the worst case SET pulse width taking into account the PIPB effects and so on. At the end, the zero-timing SET mitigation algorithm is executed to generate the final SET mitigated design. Please notice that the SET mitigation step can be carried out with extra user constraints declaring specifically to include or exclude certain part of the design for charge sharing structure insertion. Figure 4.9 is representing the pseudo code of the developed algorithm.

The goal of the mitigation algorithm is insertion of extra charge sharing structure into the design to reduce the possible SET pulse amplitude and width when it traverse the logic paths. Since the insertion of the charge sharing structure increases the fan-out of the selected nodes, the algorithm also controls the threshold for insertion of the fan-out to avoid introduction of extra delay in the path which results in performance

degradation. This is possible due to lower buffering threshold level provided in FPGA routing node: if the fan-out is below that threshold, the delay of the traversing signal is not affected. On the other side, the added charge sharing structure will reduce the PIPB effect which leads to nullifying the SET effect before reaching a memory element such as Latch, Flip-Flops or IO blocks.

The mitigation algorithm starts with loading the Post-Layout Netlist and the SET report generated by the SETA tool. The mitigation is performed in three phases:

1. It computes the Resistive Capacitive(RC)load for each circuit node within the original netlist. The computation is done adding the timing of buffered or un-buffered nets connected to the output pins of the considered node. The coefficient of the RC load is obtained dividing the timing by the fan-out of each node.
2. It selects the suitable nodes for the logic charge sharing insertion. The selection is done by interpolating the PIPB values considering the original RC load and obtaining the expected size of charge sharing structure for each node in terms of number of gates. The selection of the node where to apply charge sharing gates is determined by combinational permutation identifying the solution that minimize the PIPB effect while limiting overall number of added gates per circuit logical path. Figure 4.10 is reporting an example of interpolation data, where it is possible to notice that when charge sharing structure contains less than 40 gates, it is possible to achieve a reduction of the PIPB coefficient, for example filtering of SET without affecting the timing characteristics of the node.
3. It modifies the netlist adding charge-sharing structure of proper size to the selected logic nodes. Finally, the algorithm exports the modified netlist and placement constraints file. The two files then can be imported in the commercial FPGA design flow to generate the final design implementation. An example of the application of charge sharing mitigation algorithm is shown in Figure 4.11. To elaborate more, considering the original netlist A where three SETs pulses having width of 0.3, 0.6 and 0.8ns can be broadened up to 0.38, 0.81 and 1.08 ns since PIPB coefficient of all gates are positive. Considering the application of the charge sharing structure at the netlist B, an electrical masking or reduction of all the SET pulses is noticeable.

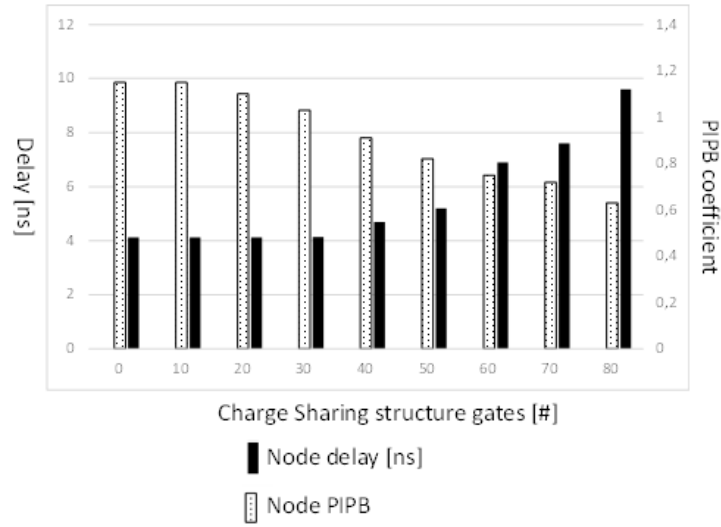


Fig. 4.10 Charge sharing number of gates per logic nodes with respect to the routing delay and PIPB coefficient.

Table 4.3 Characteristics of the original benchmark circuits

Circuit	Versatile[#]	FFs[#]	frequency[MHz]
B05	415	66	47
B09	493	67	46
B12	565	123	48
B13	162	50	52
CORDIC	956	240	45
RISC	1,401	1,156	42

## 4.2.2 Experimental results

In order to verify the effectiveness of the proposed mitigation technique, we select several circuits with different complexity: 4 circuits from the ITC99 benchmark collection [22], a Cordic core and a RISC microprocessor.

The selected circuits have been implemented on A3P250 Flash-based FPGA manufactured by Microsemi having 6,144 logic VersaTile. Libero SoC commercial design flow has been used in order to generate the PDC, SDF and Post-layout netlist in Verilog for evaluating the mitigation algorithm. Table 4.3 is reporting the characteristics of the selected circuits, such as number of VersaTiles configured as Logic function or FLip-FLop and the maximal working frequency.

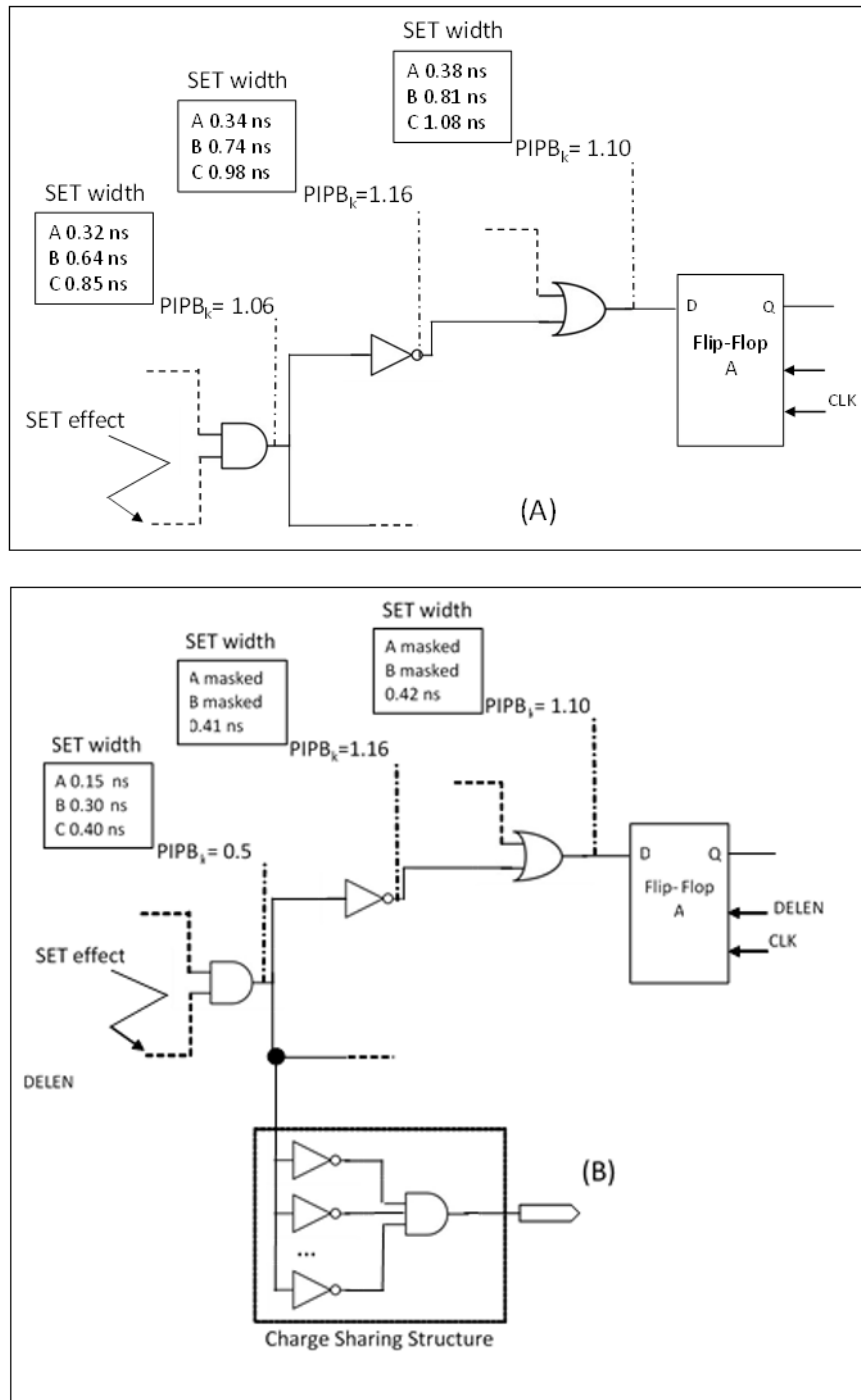


Fig. 4.11 The key concept of the Charge Sharing mitigation algorithm.

Table 4.4 Comprehensive Flip-Flop SET sensitivity using the static analysis tool

Circuit	SET width lower than 1 ns			
	Logical Masked[#]	Filtered[#]	Partially Filtered[#]	Broadened[#]
b05	46	9	3	8
b09	47	3	6	11
b12	102	1	7	13
b13	21	14	28	39
CORDIC	161	12	28	39
RISC	572	204	184	196

As it has been mentioned in the flow, we started with the classical design chain, exporting the information required to the proposed environment. We analyzed the circuit SET sensitivity using SETA tool. Since the SET width lower than 1ns corresponds to the most probable events generated by heavy ions strike on the Flash-based FPGA 130nm technology [70], as a source SET, we choose three types of SETS(0.3, 0.6 and 0.8ns). Table 4.4 is representing the results of SETA tool. (This table needs to be fixed)

The results are categorized in the following groups:

1. Logical Masked: represents a case in which no SET in the input cone could reach the FF due to logic mask.
2. Filtered: dedicated to cases in which no SET in the input cone are totally filtered.
3. Partially filtered: representing the Flip-Flops where the SET in the input cone could reach the Flip-Flops but with a reduced width.

We evaluate two versions of circuits while applying two mitigation techniques: Firstly, applying place and route and guard gate mitigation techniques using a maximal guard-gate filtering of 1ns, known as P&R-GG [71]. Secondly, our proposed method without excluding Flip-Flops in the design during SET mitigation. For both versions, Synopsys Synplify TMR has been applied exclusively on the Flip-Flops as in real design, mitigation solutions for SET would be usually adapted.

We used the electrical pulse injection platform to inject in random sensitive nodes of the circuits. We injected 5,000 SETs lower than 1ns for each circuit and report

Table 4.5 SET fault injection wrong answers comparison

Circuit	Wrong Answer[%]		
	Plain	P&R-GG	Proposed Method
b05	68.5	12.2	4.3
b09	72.6	8.4	2.6
b12	83.2	9.4	3.1
b13	54.8	16.5	4.1
CORDIC	89.4	19.6	4.3
RISC	94.6	21.6	4.8

Table 4.6 Timing and area overhead for each method

Circuit		b05	b09	b12	b13	CORDIC	RISC
Timing[%]	P&R-GG	12	13	15	16	19	18
	Proposed Method	0	0	0	0	0	0
Area[#]	P&R-GG	27	28	28	27	32	31
	Proposed Method	25	27	25	24	28	27

the results in Table 4.5 where we show the percentage of *wrong answers*, when the circuit produces at least one output data different from the expected one.

From the Table 4.5 it is observable that our method leads to the decrease of wrong answers drastically with a noticeable improvement with respect to the result reported at [71]. Table 4.6 reports the timing area overhead in terms of max frequency degradation percentage and number of VersaTiles respectively, of the two SET mitigation solutions against original version. It is noticeable that our proposed method provides no timing penalties. Moreover, the area overhead is slightly less than the P&R-GG method.

### 4.2.3 Research advancement on mitigation of Single Event Transient

In order to tolerate the implemented design regarding SET phenomenon, two mitigation solutions have been proposed. The mitigation solution based on inserting Guard Gate logic block and mitigation solution based on inserting charge sharing logic gates. In the first approach, inserting Guard Gate logic block, the SET pulse is filtered by passing through the logic blocks. There are several approaches that apply the same mitigation solution and applying the filtering blocks before all the storage el-

ement of the implemented circuits. Taking as an example the new radiation hardened RTG4 Flash-based FPGA, which provides the possibilities to add the Guard Gate logic before each Flip-Flops on the implemented circuits. However, the proposed approach has been known as the first one that is not applying the filtering blocks before all the storage elements but adding the blocks in the sensitive Flip-Flops of the implemented design. Moreover, using the proposed workflow, it is possible to tune the filtering capability with respect to the duration of the SET pulse. However, using this methodology, the increasing of the area overheads and timing penalties are an open issues. Therefore, the second mitigation has been proposed. The second methodology which is based on the charge sharing concept, is known as the first methodology able to filter SET pulses without introducing any timing penalties.

# Chapter 5

## Industrial Application

### 5.1 Radiation Test: Ultra High Energy Heavy Ion Test Beam on Xilinx Kintex-7 SRAM-based FPGA

During recent years, FPGAs have attract more attentions due to their increasing performances and high flexibility. However, in order to apply theses devices to mission critical applications such as avionic and space missions applications, the reliability of FPGA device has to be evaluated against faults and errors induced by radiation effects. Regarding SRAM-based FPGA, the SRAM cells holding the configuration data of the circuit design implemented on the device are among the devices highly susceptible against Single Event Upsets (SEU) induced by radiation effect when hit by charged particles [72]. One SEU in the configuration memory can cause a system misbehavior depending on the affected bit. Therefore, it is necessary to evaluate the device and design sensitivity against SEU in configuration memory and apply suitable fault tolerant strategy to reach a successful mission. Radiation test is one of the common methods used for such evaluation since it provides the most similar environment with respect to real space environment. Radiation test can emulate the space environment using accelerated particles to apply to the Device Under Test (DUT). Considering the necessity of evaluating the device behavior in the radiation environment, we performed the radiation test on a Xilinx Kintex7 SRAM-based FGPA device using first-ever availbe Ultra High Energy(UHE) HI beam, defined as ions in the 5-150 GeV/n range, provided in CERN.



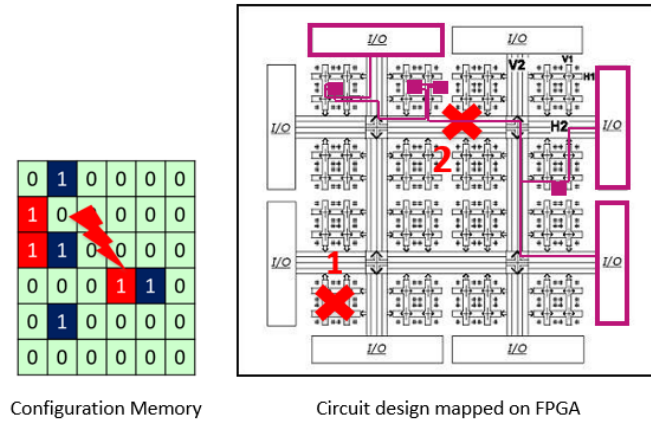


Fig. 5.1 SEU in configuration memory may corrupt circuit design mapped on FPGA

## 5.2 Background

The high sensitivity of SRAM cells against SEE induced by radiation effects with respect to its important role in SRAM-based FPGA, SEUs in configuration memory are becoming one of the main reasons of errors and misbehavior. As it is represented in Figure 5.1, one SEU (bitflip) in the configuration memory, depending where such bit is used to configure the resources in devices such as Look Up Table (LUT) and Programmable Interconnection Point (PIP), may corrupt the circuit implemented on the device.

There are many methods for mitigating SEUs in configuration memory, such as traditional redundancy based solutions and various configuration memory scrubbing techniques. Triple Modular Redundancy (TMR) as one of the most popular fault tolerant techniques has numerous implementations and variations. As for SRAM-based FPGA manufactured by Xilinx, a Xilinx TMR tool [73] is available to implement TMR automatically on a design which triplicates the logic paths and other sequential elements including Flip Flops (FFs) and block memories together with automatic voters insertions, as it is shown in Figure 5.2.

The XTMR implementation provides a fine granularity SET mitigation in configuration memory as long as two replicates of same logic path segment are affected by SEUs in configuration memory, the circuit design is still able to carry out normal function. For example, in Figure 5.2, if location 1 and location 2 are affected by SEUs simultaneously, the design works correctly. On the other side, if location 2 and

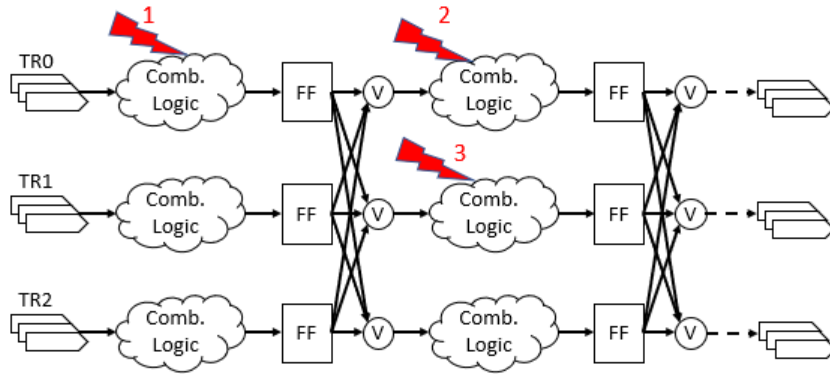


Fig. 5.2 SEUs in configuration memory affects different copies of logic path in XTMR implementation

location 3 are affected by SEUs, the design generates wrong results for the segment which may propagate through the while circuit design causing system misbehavior.

The second traditional methodology is known as configuration memory scrubbing. This methodology is refreshing or rewriting the configuration memory with the correct bitstream periodically or triggered by some detection mechanism. The important point of these technique is to define when and how the rewriting is performed in a way to not interrupt the function of the design frequently. For example, to not interrupt the functionality of the design when the error rate cross-section could satisfy the reliability requirement posted by application constraints.

Several strategies have been used for triggering configuration memory scrubbing. The most common one is blind scrubbing which is not involved with any error detection mechanism. Therefore, it rewrites the configuration memory periodically and it is critical to determine the frequency of the scrubbing which should be calculated relying on the error rate cross-section data of the device and design in the target environment. More sophistic and complicated scrubbing techniques exploit other features of the devices and circuit design itself, for example in [74], the scrubbing is done in frame instead if the whole configuration memory by exploiting the partial reconfiguration capability of the device. In [75], scrubbing is scheduled according to the criticality of the hardware task so that a better trade-off between the system reliability and scrubbing overhead could be achieved. Therefore, for optimizing the scrubbing rate for reducing system availability overhead, error detection of SEU in configuration memory should be applied. One simple error rate detection mechanism could be applied by exploiting features such as the Frame Error Correcting Code

(ECC) and Soft Error Mitigation (SEM) IP provided by Xilinx for its SRAM-based FPGA devices. The error rate cross-section data of DUT in the target environment is highly beneficial for better trade-off of performance and desired system reliability against SEE induced by radiation effects. Radiation test as one of the methods for gathering the error rate cross-section data are quite popular among academic and commercial projects for evaluating different devices and designs to be deployed in space applications.

Several radiation test have been performed and the data of these radiation test are available in literature with different types of electronic devices and different radiation beam recipes. Regarding SRAM-based FPGA devices, several radiation test have been performed. For example, [76] reports the test results performed on Xilinx Virtex-5 FPGA under both neutron and proton radiation beam in different facilities. Xilinx Kintex-7 FPGA has been tested under proton beam and the results are reported in [77]. [74] is reporting the data of the radiation test performed on Xilinx Virtex-5 FPGA under neutron beam evaluating a Frame-Level Redundancy Scrubbing (FLR-scrubbing) techniques.

### 5.3 Device and Design Under the Test

We performed a radiation test on Xilinx Kintex-7 SRAM-based FPGA using Ultra High Energy heavy ion test beam for the first time available at the radiation center of CERN.

During the radiation test, a Xilinx Kintex7 FPGA KC705 Evaluation Kit equipped with a Kintex7 XC7K325T SRAM-based FPGA was used as DUT. The same as radiation test that had been performed on Xilinx Virtex5 [76], an ARM-based SoC, as illustrated in Figure 5.3, was used as benchmark circuit which contains an ARM Cortex-M0 processor provided by ARM as flattened netlist through University Program, a UART peripheral as input and output device, a block memory implementation via Xilinx BRAM IP for holding software code and data and a clock generator to convert the differential clock source on board to a single end clock signal in system. The UART and BRAM component are attached to an AHB-Lite bus as same as Cortex-M0 processor.

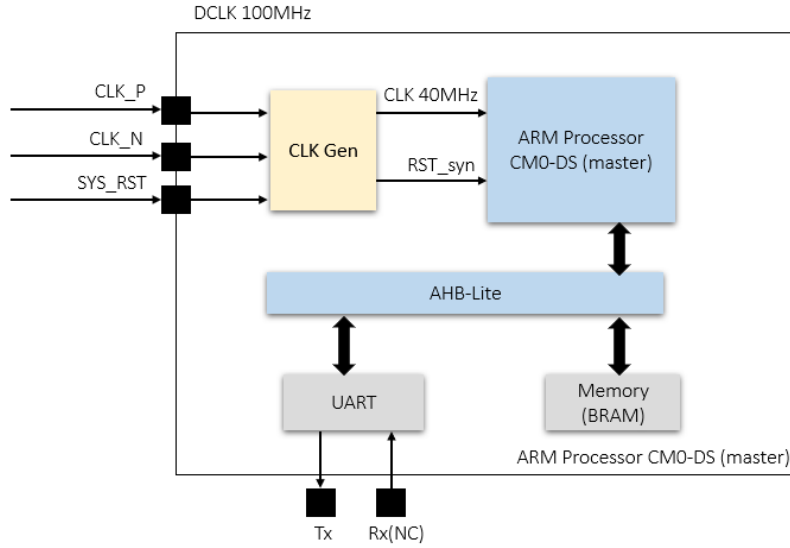


Fig. 5.3 Original ARM-based SoC used as benchmark circuit

However, with respect to the Virtex-5 device used in previous radiation test, Kintex-7 has much larger amount of resources [78]. Therefore, in order to maintain the utilization of the resource to capture as many effects as possible during the test, the ARM based SoC was replicated multiple times which is represented in Figure 5.4.

Two version of the ARM-based SoC have been implemented:

1. Plain version: the original ARM-based SoC replicated for 50 copies on kintex7 device.
2. XTMR version: based on Plain version with Xilinx TMR applied, replicated for 10 copies.

Table 5.1 is reporting the hardware utilization for both versions. As it is noticeable, the resource overhead by XTMR implementation is as high as 400%. This overhead is because of the extra voters inserted and more complicated routing caused by three replicated logic paths and voter connections.

Please notice that  $*\_SoC$  is the utilization data for one copy in the final design,  $*\_x50$  and  $\_x10$  means the original ARM SoC is replicated 50 and 10 times for Plain and XTMR version respectively.  $^+$ Utilization of LUTs for each copy in the final design has a small difference among them due to later place and route stage.

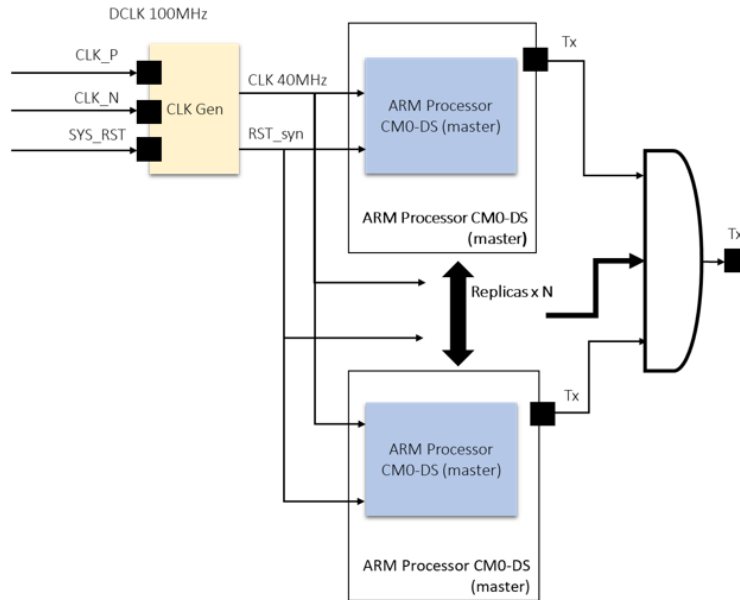


Fig. 5.4 Replication scheme of ARM-based SoC for increasing device utilization

Table 5.1 Resource utilization for Plain and XTMR Version of ARM-based SoC on Kintex-7

Version	LUT[#]	LUT[%]	FF[#]	FF[%]	BRAM[#]	BRAM[%]
<i>plain_SoC</i> *	~ <sup>+</sup> 3.907	~ <sup>+</sup> 1.91	1.189	0.29	4	0.89
<i>plain_x50</i> **	195.074	95.72	59.460	14.59	200	44.94
<i>XTMR_SoC</i> *	~ <sup>+</sup> 18.760	~ <sup>+</sup> 9.20	9.057	2.22	12	2.70
<i>XTMR_x10</i> **	187.557	92.03	90.572	22.22	120	26.97

As it is presented in Figure 5.4, since the number of pins that can be used and monitored during the radiation test is limited, for the last design in DUT, the plain and the XTMR version have their replicas outputs ANDed together to reduce the number of pins in the final design. Therefore, as long as one of the replica generated an error in the output signal, the error will propagate to the output of the top-level design and captured by the monitor during the test.

For the program running on the ARM processor, a bubble sort application was implemented to generate the ascending and descending sorted results of a pre-defined array in the code, which were sent to UART component in the design as output. The application is executed in a deadloop to continuously generate output to be monitored from outside.

## 5.4 Monitoring setup

Apart of Kintex-7 board, Zybo board [79] is used to monitor the DUT outputs along with a custom designed Host PC application. After the initialization of the test, the Host PC application will program the monitor board which is the Zybo board to initialize the UART component to continuously monitor the outputs from DUT and program DUT to start the run loop. During the run time, the monitor board keeps checking the UART outputs from DUT and compare them with the golden copy captured and stored before radiation test (fault free output). In the case of a mismatch, Host PC application will be notified by monitor board and starts the configuration memory readback procedure of DUT and after the UART log data transferred from monitor board to Host PC and stored, the UART monitors in monitor board are reset and a new run starts as represented in Figure 5.5. During each run, the Kintex-7 DUT is reprogrammed to clear out any SEU in configuration memory accumulated in previous run.

## 5.5 UHE Heavy Ion beam

The Xe heavy ion beam is used for the radiation test with the energy level set to  $40 \frac{GeV}{n}$  and the effective LET is  $3.7 \frac{MeV*cm^2}{mg}$  obtained by FLUKA [80] while considering the volume around  $1\mu m^3$ . The particles with such high energy level is capable to

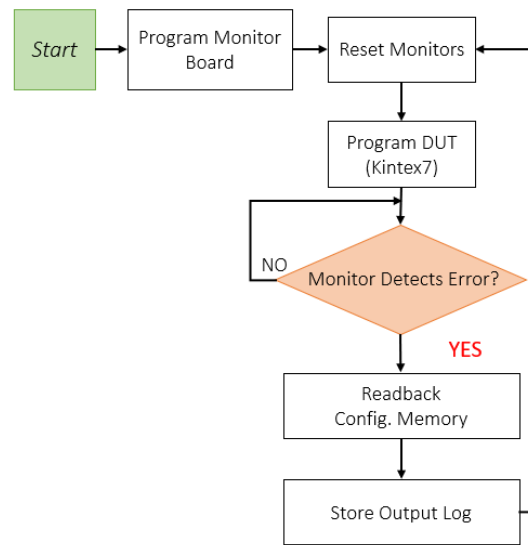


Fig. 5.5 Monitor flow with the Host PC application

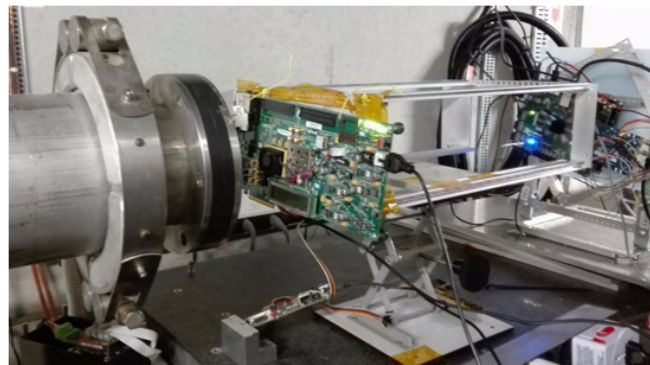


Fig. 5.6 board and beam setup for alignment.

penetrate the device with package, and possible to generate Single Event Multiple Upsets(SEMU) in the configuration memory.

A logbook containing the beam information regarding number of particles hitting the devices along with timestamps during the radiation test is provided so that it is possible to correlate the error detected during the test and number of particles afterward for calculating the error rate cross-section. The board and beam setup has been shown in Figure 5.6.

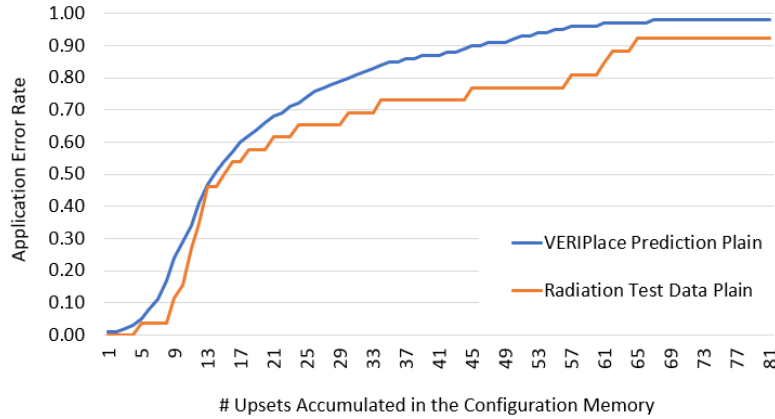


Fig. 5.7 VERIPlace error rate comparison with radiation test data for plain version

## 5.6 Radiation Test Data

### 5.6.1 Error rate analysis

In order to provide the Error Rate analysis, we used a tool developed in house [81], which is responsible for performing design application error rate prediction for SRAM-based FPGA with respect to SEU in configuration memory induced by radiation effects and improvement of design reliability without introducing hardware resources overhead. The hardening techniques used in this methodology has been verified considering the previous radiation test that has been performed in our group for prediction of error rate [76].

The application error rate is defined as the probability of application generating an error at the output with respect to the certain number of SEU accumulated in the configuration memory. Figure 5.7 is representing the comparison between the radiation test data and VERI-Place prediction for Plain version while Figure 5.8 is dedicated to the comparison for the XTMR version.

As it can be concluded from Figures 5.7 and 5.8, the prediction made by the VERI-Place tool is accurate. The minor observed offset is due to the factor that the high energy particles hitting the device may generate SEMU and the beam provided at CERN was operating as spilling mode. To elaborate more, the beam was provided periodically as a burst of particles is directed to strike the device. Therefore, it leads to over counting of SEUs in configuration memory during the radiation test.



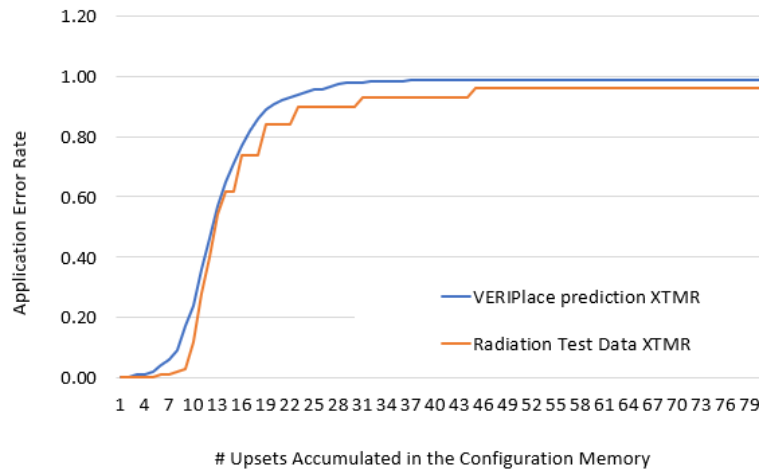


Fig. 5.8 VERIPlace error rate comparison with radiation test data for XTMR version

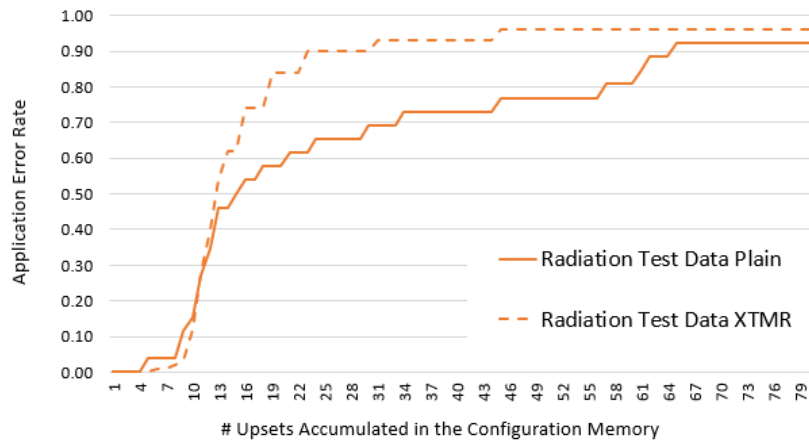


Fig. 5.9 VERIPlace error rate comparison between the Plain and XTMR version collected during radiation test

As a next step, we compared the radiation test data of a Plain and XTMR versions. As it can be observed in Figure 5.9, XTMR has lower error rate when number of SEUs accumulated in the configuration memory is relatively low. However, when the number of SEUs increases, the error rate of XTMR version goes higher than the plain version. This result is reasoned by large resource overhead in XTMR version which can provide larger sensitive area or bits in configuration memory. On the other side, it shows that when XTMR is complimented with other techniques such as configuration memory scrubbing to avoid SEU accumulation in configuration memory, XTMR version can achieve quite low error rate.

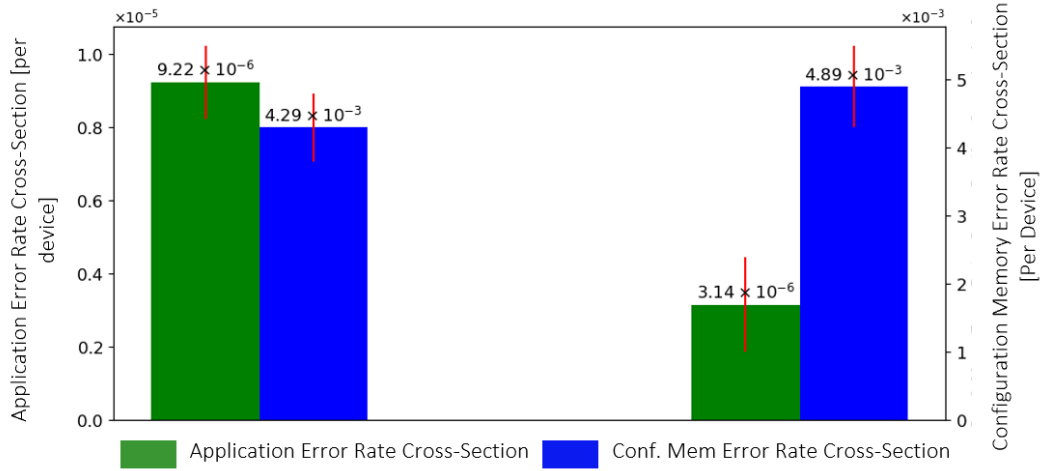


Fig. 5.10 Application and configuration memory error-rate cross-section comparison for Plain and XTMR versions.

Moreover, the application error rate cross-section defined as the probability of a particle striking the device generating an error in the output, together with the configuration memory error rate cross-section which is the probability of a particle striking the device generates a SEU in the configuration memory are reported in Figure 5.10.

The application error rate cross-section depends on the circuit design and in this case the software running on the soft-core Cortex-M0 processor. On the other hand, the configuration memory error rate cross-section is dependent to the device. Therefore, as it is represented in figure 5.10, it is similar between the Plain and XTMR version. Also XTMR version is able to achieve the application error rate cross-section 65.9% lower than the Plain version.

## 5.6.2 Observation of SEMU

After the radiation test, readback data files of the configuration memory were collected for each for Plain and also XTMR version. From the binary readback file, frame data has been extracted in order to find the actual SEUs occurred in the configuration memory during the test. During our test, we focused on the configuration memory. Therefore, the mask file was used to remove the dynamic content. For example, Block RAM in the readback file which is described in details in [82].

By analyzing the location and position of SEUs in the configuration memory, several patterns have been recognized as multiple upsets occurred close to each other forming a cluster which is providing the possibility to be an actual SEMU occurrences.

Two bit in the configuration memory are defined as close when they reside in the same major column and the distance is calculated following equation 5.1 is less than  $\sqrt{2}$ . Similar Multiple Bit Upset(SMBU) have been seen in reports of previous radiation test with lower energy [83], [84].

$$dist(a,b) = \sqrt{(LFA(a) - LFA(b))^2 + (BitOffset(a) + BitOffset(b))^2} \quad (5.1)$$

In this equation, LFA is the Linear Frame Address in readback data while BitOffset is the bit offset within the frame.

Figure 5.11 is representing the pattern found by analyzing readback file. As it can be concluded, event though the size of cluster can go up to 6 and up to 3 adjacent bits in the same frame may be corrupted at the same time, no cluster across three frames has been observed while Figure 5.12 is representing the distribution of the clusters of different sizes including the isolated bitflip for example cluster size 1 for both Plain and XTMR version.

The cross-section is calculated as number of clusters of certain sized divided by the number of particles passed through the device across all the runs for both Plain and XTMR version during the radiation test. Table 5.2 is reporting the previous radiation test data [84], which has been performed at the same LET as this test but with Si ion beam. However, the Xe ion beam has much higher LET leading to much higher bitflip cross-section.

In our experiment, we observed cluster for larger size such as size 5 and 6 which is not observed in the experiment with lower energy level heavy ion beam. Comparing with Si beam, distribution of larger cluster is higher with the Xe UHE beam even though LET is lower. Considering Xe beam with lower energy, it is important to notice the UHE beam LET is much lower. Therefore, UHE beam present characteristics regarding SEMU effect which may not be trivial using lower energy beam for accelerated radiation test and unfold data later for GCR spectra for example the result from UHE beam could be closer to real scenario when application under GCR radiation environment is considered.

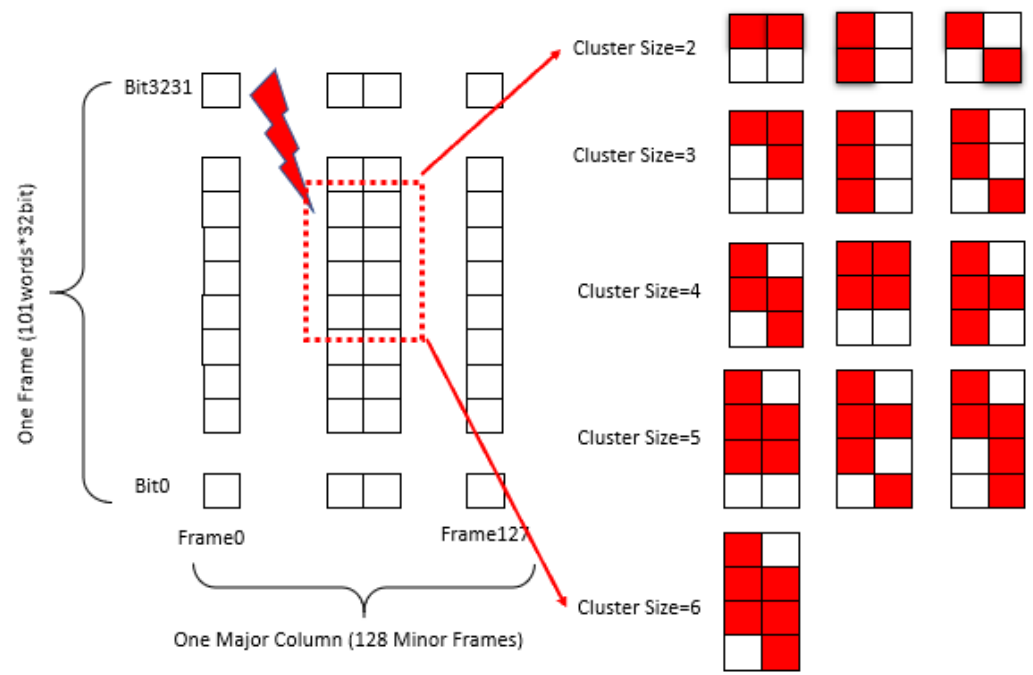


Fig. 5.11 Cluster (SEMU) patterns observed during radiation test.

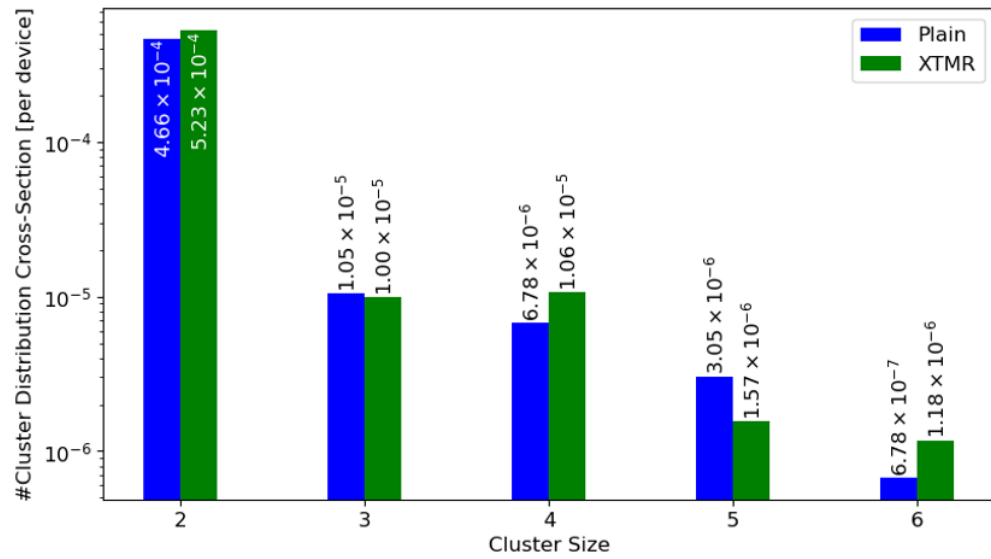


Fig. 5.12 Cluster distribution cross-section of different cluster sizes.

Table 5.2 Comparison with Test Result of Lower Energy Beam on Kintex-7

	$Si^*$	$Xe^*$	Xe(UHE)
LET ( $\frac{MeV.cm^2}{mg}$ )	4.35	49.3	3.7
CMem Bitflip Cross Section ( $\frac{\#bit\ flip}{particle}$ )	$3.8 \times 10^{-1}$	$4.67 \times 10^{-3}$	
% Cluster Size=1	90.1	64.0	88.66
% Cluster Size=2	8.7	23.0	10.85
% Cluster Size=3	0.6	4.1	0.22
% Cluster Size=4	0.2	3	0.20
% Cluster Size=5	-	-	0.05
% Cluster Size=6	-	-	0.02

Evaluating the occurrence of SEMU(cluster) and the cross-section(Probability) is critical for analyzing system reliability against SEEs in configuration memory induced by radiation effect. More importantly, when certain fault tolerant technique is to be applied and evaluated. For example, the configuration memory scrubbing methods [85], [86] and ECC based techniques such as built-in FrameECC in Xilinx Kintex-7 FPGA devices [74], may not be sufficient alone with large clusters occurs corrupting multiple hits across different frame.

Considering XTMR solution, please notice that the SEMU poses further obstacles as it the place and route of three logic path replications did not take into account the possibility of SEMU, there is a chance two bits in the configuration memory close to each other control two of the logic paths of the same segment, which means one single particle hitting the device leads the cluster corrupt which cause the circuit design to face an error. The developed VERI-Place tool is able to improve the design reliability, by modifying the place and route of design. Therefore, the final configuration memory contains reduced number of sensitive bits considering also SEMU. To summary, we performed a radiation test on Xilinx Kintex-7 SRAM-based FPGA using UHE heavy ion beams available at CERN. The error rate analysis and comparison with error rate prediction performed by the VERI-Place tool shows that the XTMR version of design is able to achieve a well reduced sensitivity against SEUs in configuration memory induced by radiation effects. The overall application error rate cross-section of XTMR is 65.9% lower than the plain version.

Moreover, SEMUs have been observed as clusters of different sizes in the configuration memory readback file, which means further actions may need to be taken to cope with the possibility of multiple upsets in configuration memory corrupting

multiple resources in the circuit design at the same time, for instance two logic path replicas in XTMR implementation.

## 5.7 EUCLID Space Mission

### 5.7.1 What is EUCLID?

EUCLID is a cosmology mission with the goal to study the geometry and the nature of the dark universe, dark matter and dark energy. The mission will investigate the distance-redshift relation and the evolution of the cosmic structures by measuring shapes and redshift of distant galaxies. EUCLID space segment will be spacecraft placed into an orbit around L2 (around 1.5 million kilometers from earth) with a coverage of  $15,000 \text{ deg}^2$  in 6.25 years with step and stare observation strategy. The launch of the spacecraft is planned for 2020. EUCLID spacecraft will host 2 instruments:

1. Near Infrared Spectrometer Photometer (NISP)
2. VISible Imager (VSI)

Both instruments take advantages of functions implemented in FPGA devices while the controller part of the units are adopting RTAX devices, the elaboration part is essentially adopting Radiation Tolerant ProASIC Flash-based FPGA technology which are widely used in space application in both NASA and ESA mission because of the high level of immunity to radiation effects. For this device family in particular, the technology is guaranteed to:

- Single Event Latchup (SEL) immune up to  $68 \text{ MeV/cm}^2/\text{mg}$
- Total Ionizing Dose (TID) better than 30 Krad
- F/F with TMR
- No SEU effects are expected in the configuration memory

Considering these features, we identify a radiation profile that has been characterized as the maximum exposure of 4krad which leads to the generation of SET pulses

with duration between 0.43 ns and 0.52 ns. In order to perform radiation hardening of FPGA design in ProASIC, several steps have been identified:

- Radiation Hardening at RTL level
- Radiation Hardening of the I/Os
- TMR of F/Fs
- Static Timing Analysis (STA)
- Applying SETA flow

Considering the mentioned steps, the radiation hardened steps starts from the original netlist. The netlist has been implemented using VHDL while TMR has been applied to all the Flip-Flops. The netlist with the TMR Flip-Flops passed through place and route performed with Microsemi Designer. As a next phase, Static Timing Analysis(STA) has been performed. By performing the functional verification with test-bench on the Post-layout netlist, the netlist has been provided to SETA flow in order to perform the SET analysis and mitigation based on the performed analysis.

The EDA tool that we developed named as SETA and applied to EUCLID space mission project is represented in Figure 5.13.

The flow that is illustrated in Figure 5.13 has been applied to EUCLID netlist. The iteration showed in the Figure 5.13 has been performed several times. To elaborate more, we need to run the tool for several iterations to overcome timing failure and fulfill the timing requirements. The tool starts with the commercial tool which in this case it is and implemented Microsemi Libero SoC 11.8 and extract the Post-layout Verilog netlist, AFL and PDC files. From this information, as a first iteration, SETA evaluates the impact of SETs on the circuit functionality by calculating the SET propagation in all the circuit nodes and the maximal SET pulse width at the input of each Flip-Flop. This information is reported in the *SET report* from SET analysis. This report is provided to mitigation tool which is focusing on the filtering of the SET reaching to the Flip-Flops by inserting Guard-Gates(GGs). The following concept is classified in three phases: The First phase reporting the SET sensitivity analysis of EUCLID original netlist. The second phase is for elaboration of the steps for mitigating the netlist. The third phase is dedicated to SET analysis of the mitigated netlist.

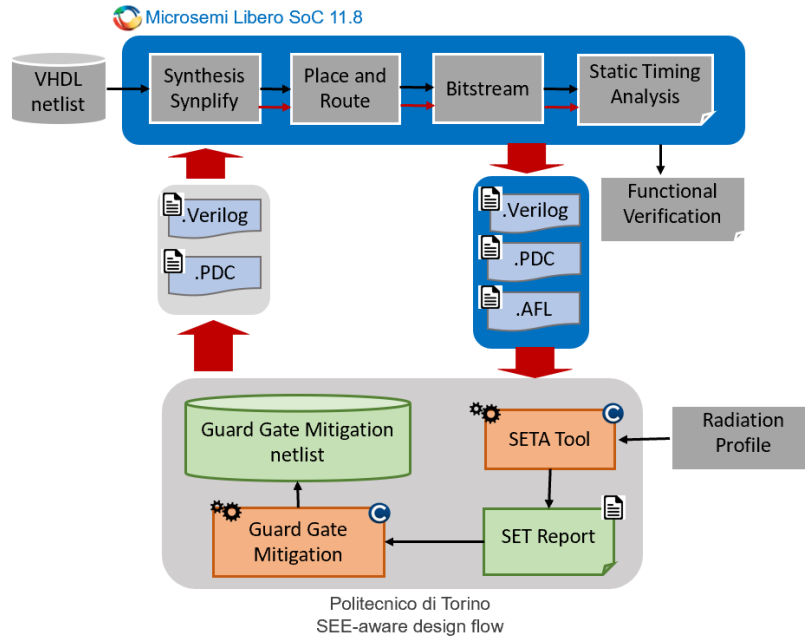


Fig. 5.13 The EDA adapted flow integrates both commercial tool (Microsemi Libero Soc 11.8) and the SET analysis and mitigation flow.

Table 5.3 Circuit Resources of the EUCLID netlist by type

Type	Core Tiles[#]
Combinational Logic	30,190
Sequential Elements(Flip-Flops)	17,718

### 5.7.2 Analysis of EUCLID original EUCLID netlist sensitivity to SET

The EUCLID netlist has been provided to the developed EDA tool. Table 5.3 is reporting the resource usage of the EUCLID netlist while Table 5.4 is reporting the timing characteristics.

In Table 5.4, the component *CLK\_60M*, *CLK\_20M*, *CLK\_60M\_buff* and *CLK\_30M* are the components in the design corresponding to different clock domains whose frequencies are as is it represented in the name, while *SPW\_CTLR0* and *SPW\_CTRL1* are two SpaceWire controllers implemented in the design.

Considering the radiation profile features, it has been defined that the expected SET pulses have width equal to 0.519ns, 0.488ns, 0.462ns and 0.437ns. These SET pulses have been provided to SETA as inputs in order to perform a SET sensitivity



Table 5.4 Timing resources of the EUCLID netlist

Reference name	Period[ns]	Frequency[MHz]
<i>CLK_60M</i>	12.097	82.665
<i>CLK_20M</i>	30.156	33.161
<i>SPW_CTLR0</i>	11.177	89.469
<i>SPW_CTRL1</i>	13.437	74.421
<i>CLK_60M_buff</i>	12.097	82.665
<i>CLK_30M</i>	28.022	35.686

Table 5.5 Single Event Transient Analysis for SET Ranging from 0.43 ns to 0.52 ns representing the number of Flip-Flops for each case

Source SET[ns]	Totally Filtered[#]	Partially Filtered[#]	Broadened[#]
0.520	11,130	0	6,542
0.488	11,130	0	6,542
0.4462	11,130	0	6,542
0.437	11,162	6,510	0

analysis and report the sensitive node location. Table 5.5 is reporting the result of SETA tool regarding the sensitivity of EUCLID design. In this table, *Totally Filtered* is reporting the number of Flip-Flops which are facing the SET pulses that during their propagation, the pulses have been totally filtered. Please note that these number of Flip-Flops is not including the Flip-Flops implemented for I/Os; *Partially Filtered* is dedicated to SET pulses propagating up to Flip-Flops while the width of the pulses have been reduced but not completely filtered which leads to lower probability of the pulses being sampled; *broadened* is representing the pulses that during the propagation, the width of the pulses have been increased during the propagation. This case is introducing the most critical situation since by increasing the width of the pulse, the probability of the pulse being sampled by the storage element is increasing.

The computational time required for performing SET analysis considering the three input SET pulses is approximately 65 hours. Please notice that, firstly, the resource used area on A3P3000RT is 63.65% and the required time depends on the computer performance that has been used for executing the tool. In our case, we used a VirtualBox machine on the mid-range laptop PC, while considering running the tool on a more powerful workstation/server, the computational time should be able to shorten more.

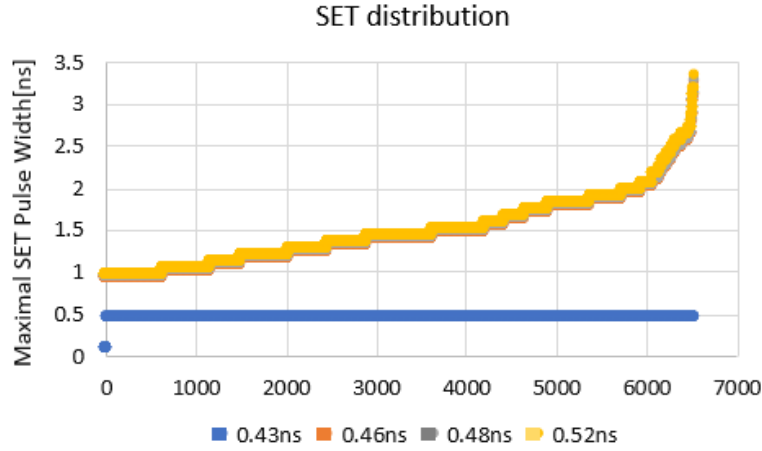


Fig. 5.14 The SET distribution obtained on the original EUCLID netlist.

The SET distribution is reported in Figure 5.14. In this figure, the horizontal axis is representing the ID of the Flip-Flops in the original netlist that are partially filtered or broadened, while the vertical axis shows the maximal SET pulse width reaching the corresponding Flip-Flops. It can be observed from the Figure that for SET pulses with short duration such as 0.43ns, the source SETs have been electrically filtered before or they are not broadened while propagating through the logics and routing nets of the circuit, before reaching to the Flip-Flops. It means that the SET pulse reaching to the Flip-Flops are almost the same width as the source SETs. This phenomena is dependent to the used technology and its behavior regarding SET pulse.

Differently, when the source SET pulses have longer duration such as 0.52ns, the SET pulses are facing a drastic increasing of the width of SET pulses during the propagation of the pulse through the circuits. This means that the probability of SET pulses being sampled by the storage elements is increasing which leads to more critical condition for the mission.

### 5.7.3 Mitigating the EUCLID design netlist

By performing the SET analysis and identifying the sensitive nodes of the EUCLID circuits, the report of SET sensitivity and the netlist have been provided to the mitigation tool. Based on these two files, the mitigation tool inserts the G logic gates

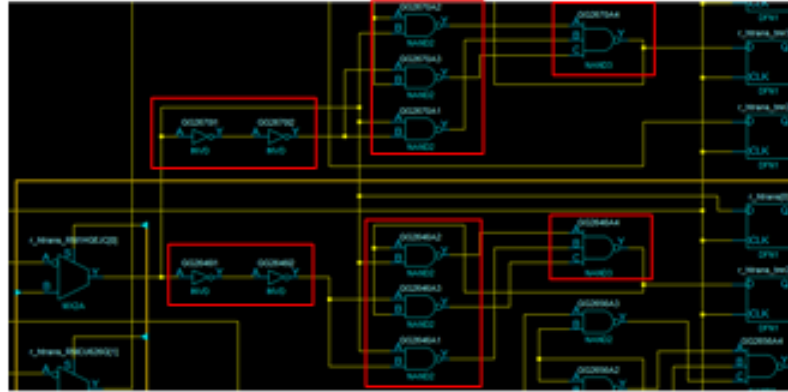


Fig. 5.15 An example of Gaurd-Gate automation insertion on a portion of the EUCLID design.

for filtering the SETs. Figure 5.15 is illustrating an example of GG insertion within the EUCLID design.

In order to mitigate the EUCLID design, based on the analysis report, we set the filtering capability of Guard-Gate mitigation to 1.350ns. By this value, we provide the maximal broadening reduction and filter all the SET pulses reported as by the analysis tool before reaching to the Flip-Flops. Although this filtering capability is efficient for removing all propagated SETs, it does not fulfill timing requirements regarding 30 MHz clock domain and the timing closure for this domain fails. Therefore, as a next step, we decrease the Guard-Gate mitigation insertion to 1.2ns. However, the timing closure regarding 30MHz clock domain fails again. As the last step, we decrease the Guard-Gate mitigation insertion down to 1.0ns maximal broadening reduction. As a result we success the timing closure for all clock domains. Table 5.6 reports the three iterations of Guard Gate mitigation tool, implementation and timing analysis. Moreover, Table 5.7 reports the area overhead of these three iteration. As it can be observed from the table, using our approach, it is affecting the high frequency domains. In fact, the performance is the same as plain implementation(Original Version).It is interesting to mention that the frequency drop of the clock domain at 60MHz(CLK60) for the EUCLID TMR implementation is due to the placement and routing algorithm not implementing any specific rules as instead requested for a TMR implementation(for example redundancy domain should be placed close to each voter partition). On the other side, the EUCLID TMR+GG includes placement constraints to force the timing characteristics of each

Table 5.6 Timing analysis for three iteration of Guard-Gate mitigation tool

Reference Name	Guard Gate Delay Coefficient		
	1.4 ns	1.2 ns	1.0 ns
	Frequency[MHz]-4krad		
CLK_60M	69,845	68,304	69,793
CLK_20M	36,480	32,234	35,045
SPW_CTRL0	76,430	74,234	79,764
SPW_CTRL1	81,832	80,024	83,043
Clk_60M_buff	61,430	64,780	69,793
Clk_30M	27,640	29,550	31,248

Table 5.7 Area Over-head Report for Three Iteration of Guard Gate Mitigation Tool

Guard Gate Delay Coefficient[ns]	Area Over-head[%]
1.4	8
1.2	3.8
1.0	1.4

Guard-Gate structure. Therefore, it maintains the timing property almost equal to the original design.

Regarding the netlist usage, Table 5.8 is reporting the netlist resource usage for the last iteration. As it can be observed, a maximal number of 4 INVDs per Guard-Gate has been inserted for a total of 708 INVD.

#### 5.7.4 Analysis of EUCLID mitigated netlist sensitivity to SET phenomena

The original netlist has been evaluated regrading SET sensitivity using SETA tool. To elaborate more, we applied the SETA tool to evaluate the sensitivity of the mitigated

Table 5.8 Circuit Resources for the Mitigation Netlist

Type	Core Tile[#]
Combinational Logic	30,190
INVD gates(GG Structure)	708
NAND gates(GG Structure)	396
Total Combinational Logic	31,294
Sequential Elements(FFs)	17,718

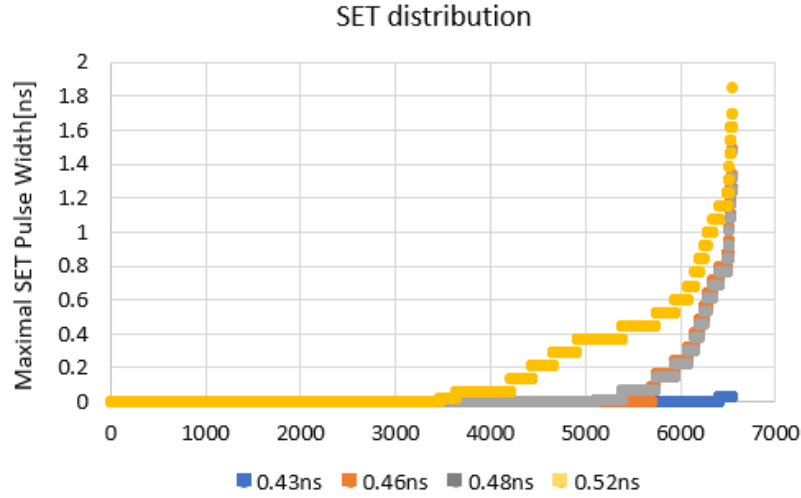


Fig. 5.16 The SET distribution obtained on the Mitigated EUCLID netlist.

circuit regarding SET pulses with the duration between 0.43 ns and 0.5 ns. Figure 5.16 illustrates the SET distribution of the mitigated netlist. As it is shown, the SET pulses below 0.4 ns will be electrically filtered before reaching the Flip-Flops inputs.

After performing sensitivity evaluation of mitigated netlist regarding SET pulse, we performed a comparison between the mitigated netlist and the original one. As it can be observed in Figure 5.17, mitigating the netlist results in removal of 97% of broadened SETs while the 3% remains with reduced pulse width around 50%. As an example, a pulse with the width of 3.2 ns in the original netlist has been partially filtered down to 1.9 ns in the mitigated version.

The CRÈME96 [87] and the developed SETA tool are applied to evaluate the EUCLID circuit SET sensitivity in terms of error cross-section for the two most resilient versions: our approach and the previous state-of-the-art solution with the TMR and SET filtering. To elaborate more, we evaluate the expected integral influence for the nominal duration of the mission as 6.25 years. For performing the SET error estimation, we normalized the CRÈME96 data for each single ProASIC3 Versatile and routing segment. The SET normalized cross-section coefficients have been elaborated with the SET tool applied to the Post-layout netlist of the target design. For the unmitigated version, we acquired a normalized transient error cross-section of  $1.77\text{E-}4$  while for the mitigated version (TMR+GG), we achieved a normalized transient error cross-section of  $2.42\text{E-}7$ .

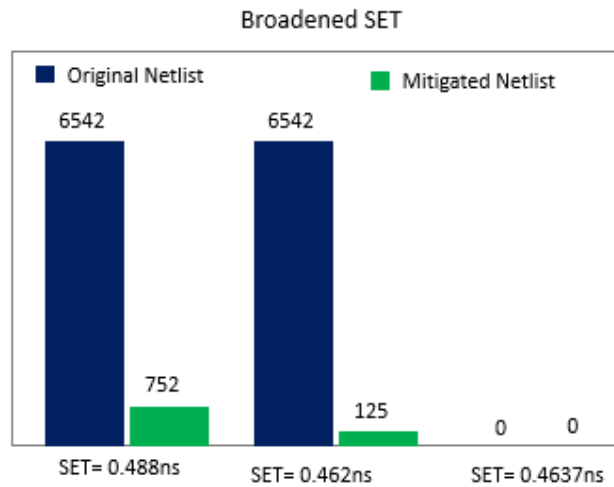


Fig. 5.17 A comparison of SET distribution between the original netlist and the mitigated netlist.

To summary, we developed a work-flow for evaluating the sensitivity of the circuits regarding the SET pulses and applying an efficient mitigation solution based on the performed analysis. This work-flow is applicable to industrial circuit as it has been applied to ESA EUCLID space mission for monitoring the dark space. This project is an industrial project carrying on by ESA and OHB with the launching plan at 2020. Experimental results demonstrated a reduction of three order of magnitude of the overall SET sensitivity. Moreover, the mitigation of SET pulses is flexible and it is possible to tuned the mitigating coefficient with respect to the design timing characteristics.

## **Part II**

### **From Transient to Permanent**





# Chapter 6

## Micro Single Event Latch-up

Single Event Effects are considered as the effect of a single particle strike in a specific location within the device which may cause different functional behaviors. Depending on the strike location, time, the electrical field and the energy of the particle, different behavior can be observed. These effects can be temporary faults that affect the device for a certain period of time, known as Soft Errors. On the other side, if the effect is permanent, it is defined as Hard Errors. One of the most critical hard errors is Single Event Latch-up (SEL) [88]. SEL is one of the major reliability concerns for VLSI devices applied to safety critical applications such as aerospace happening due to environmental radiation, includes one of the many PNPN structures of the silicon to be switched from its blocking state to a latched state, resulting in circuit malfunction [89]. Considering the reduction of circuit feature size and operating voltage level, a new kind of phenomenon has attracted attention known as Micro Single Event Latch-up ( $\mu$ SEL) [90].

SEL normally occurs near the input/output terminals of a logic gate where a charged particle can pass through the silicon device region while  $\mu$ SEL occurs at different locations across the die and between layers inducing two different effects. The first one is increasing the global current of the device. The second is the local logical stuck of the involved signals that may provoke the propagation of the faults to the circuit primary outputs. Due to the fact that  $\mu$ SEL is happening in any portion of the device, detecting this effect is a challenging issue.

Micro SEL affects depends on several aspects such as: Layout, depth, size and design density. Several studies evaluate these factors. Many works on the latch-

up effect based on the injection of transient current using Technology Computer-Aided Design (TCAD) simulation. However, due to the long simulation times, it is not applicable to large designs. 3D-TCAD simulation is used in [91] to analyze and harden SEL happening in embedded STAM-based FPGAs and a mitigation approach on the cell geometry has been proposed. To elaborate more, simulation results demonstrate that the extra source junction and source ties acquire the charge collection, leading to the higher sensitivity regarding SEL. In [92], this factor has been removed and TCAD simulation environment has been used for evaluation of this phenomenon. In [93], the effect of angular ions on the SEL cross section has been evaluated. In [94], TCAD simulation highlights the impact of both angle and roll of effects on the latch-up sensitivity showing a direct dependency of asymmetric layout considering the parasitic latch-up circuit, which is normally electrically modeled with SPIC model. Several works are dedicated to evaluate the impact of radiation strikes on the generation of micro SEL effect [7] [95] [96] [97]. MUSCA SEP3 has been developed and used for modeling the basic mechanism that occurs when an error happened due to radiation strike [97].

## 6.1 From SEL to Micro SEL

Single Event Latch-up has a permanent effect that leads to an increase of the device current. This error leads to the destruction of the device itself if not removed in time. As it is shown in Figure 6.1, besides the designed  $p$  and  $n$  transistor, additional parasitic devices that are formed by interaction of different doped area. If a current peak is injected in such a parasitic structure, because of its strong feedback net, a chain reaction is triggering, creating a short circuit between ground and  $V_{dd}$  that could burn down the device. A spurious current peak in the parasitic structures can be injected by direct or indirect ionization by means of a particle that strike the device in that area, starting the SEL effect [98]. Detection structures can be added to the sensitive transistors, in order to detect SELs and clearing them, by means of power cut off.

The latch-up current may only represent a small fraction of the normal overall integrated circuit current consumption, which leads to the creation of a localized latch-up also named Micro Single Event Latch-up ( $\mu$ SEL). In some cases, the energy deposition can cause individual cells to be unable to change state until a power cycle

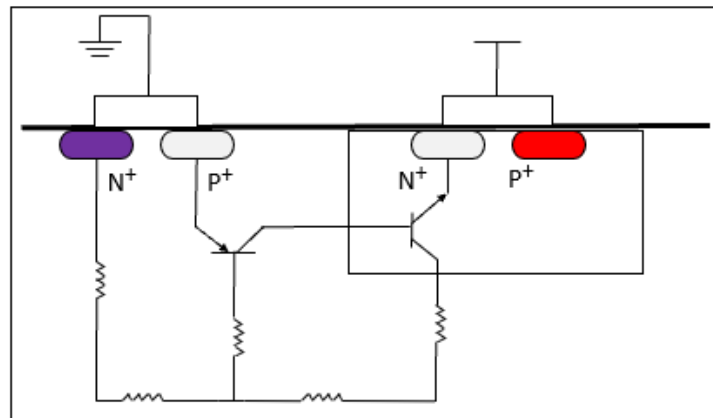


Fig. 6.1 Electrical effect generating a SEL effect

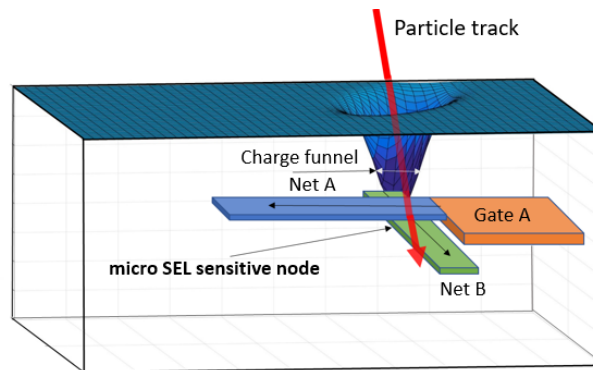


Fig. 6.2 Overview of the basic mechanisms generating a micro SEL effect on the output of a gate

is executed. These effects can cause fraction of bits to be unable to change state due to the value of collected charge [99]. These effects are normally generated by micro-dose deposition which are activated when the energy in a given region exceeds a certain threshold value.

Figure 6.2, represents the basic mechanism of  $\mu$ SEL effect. At a certain time, the particle incident cause transition of micro charge in terms of quantity that exceeds the threshold of the effect. Therefore, triggering the effect. As a result, it affects the resource within a given region, typically two nets or two cells on close metal layers [22]. Therefore, a given portion of the circuit is on a stuck logic resource leading a localized high current value.

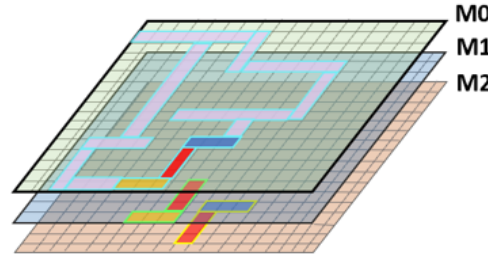


Fig. 6.3 The intra-metal layer micro-SEL effect between routing segment. The evidenced red routes represents the affected net

the physical layout of the circuit, two-ground rails conductor separates routing levels. Therefore, as a result of high charged particle interacting with the device, the overall current of the following routing section is increasing. In a case that the threshold of the current is reached, it may create a path with a low resistor values. Therefore, a temporal micro latch-up is activated. Moreover, this effect is propagating simultaneously in all the nets connected to the defected point forcing all the global net to be stuck at the logic value of one or zero, based on the used technology. Figure 6.3 represent this scenario while three routes in three different metal layers are affected.

Therefore, in order to evaluate the sensitivity of the design under the study, we develop a work flow based on the Monte Carlo method, taking into account the design under the test, placement and routing architecture within the physical layer mapping, in order to calculate the realistic micro SEL occurrences.

### 6.1.1 Micro Single Event Latch-up Analysis

In order to evaluate the sensitivity of sub-nanom micron circuit regarding  $\mu$ SEL, a methodology has been proposed which is the first methodology dedicated to geometrical layout analysis for the identification of the physical sensitive nodes and two developed engines for providing the static and dynamic error rate.

This work flow consists of five groups of tools. Figure 6.4 represents the work flow which starts with the classical IC design tool chain, going through synthesis, place and route, moving toward the developed tool for physical design description, the layer mapping and finally Monte Carlo algorithm. The combination of this

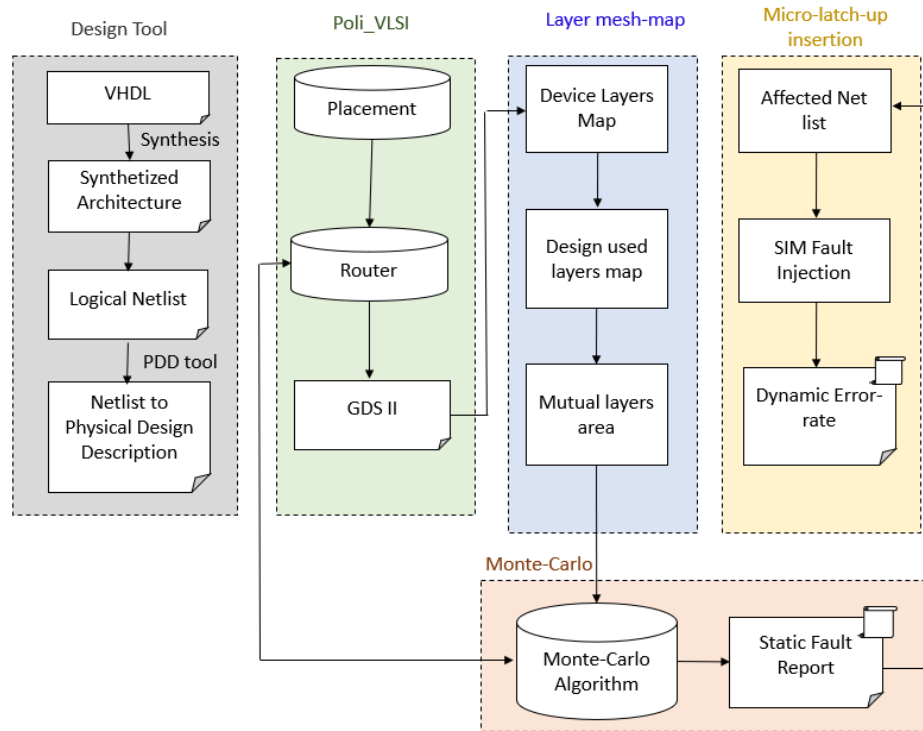


Fig. 6.4 Overview of the global analysis methodology for micro-latch up consisting on the layer mesh-map and the micro-SEL insertion tool

group of tools elaborate the layout description in order to provide the estimation of susceptibility of the design in terms of Error Rate.

The flow starts with the Hardware description of the design, VHDL of the design. The Commercial tool elaborates the VHDL in order to generate the synthesized netlist. The netlist has been used to generate the Physical Design Description(PDD) which represent the circuit graph by means of logic vertex and edge. An a next step, the placement and routing phase has been processed ny an ad-hoc physical layout too, which creates the Graphic Data System (GDS) of the final design layout. Placement has been done based on selection of the logic block location in order to provide the optimized timing features. On the other side, for routing, the shortest interconnection lines between logic elements has been selected. Therefore, GDS format for analyzing the layer mapping of the design has been generated and provided to perform and analysis on the layout.

### Micro Latch-up layout analysis

The main core of the developed work flow is the layout analysis. Layer mapping starts with generating layers meshes. A set of 2D matrix is considered for defining the layout while each 2D matrix is representing one layer of the layout. Then, the GDS of the design has been connected with the developed layer mesh, in a way that all used cubes of the GDS, has been filled to the corresponding positions in the mesh layer. Therefore, all the used cubes of the meshes layers have been extracted. The layout map contains all the placement of routing boundaries in each layer. From the layer meshes, the area of the device which has been used for routing architecture has been extracted and classified based on the layers which they belong to. If they belong to two different layers, they have been classified as a shared point between layers, which provides the possible location of micro-latchup event. Figure 6.5, represent the pseudo code of the developed algorithm.

To elaborate more, the position that are mutual between different layers have been defined. This positions can be the possible position of occurrence of  $\mu$ SEL. However, not all the radiation incident at this common points can create the  $\mu$ SEL. The radiation can cause an error only if the distance between layers are less than the defined threshold [100]. The placement of the common points who are meeting this threshold have been extracted which are adjacent enough to create  $\mu$ SEL effect with respect to the LET of the particle.

### Monte Carlo Algorithm

In order to evaluate the occurrence of  $\mu$ SEL, a Monte Carlo based algorithm is proposed and developed. The Monte Carlo algorithm generates distributions of  $\mu$ SEL within all device area available for routing interconnections. The developed environment of Monte Carlo algorithm is presented in Figure 6.6.

The generated layout mapping together with Design Netlist Description (PDD file) are imported by Monte Carlo algorithm. The Monte Carlo parameters defined by user are introduced to the algorithm in terms of the  *$\mu$ SEL Effect Rules*. All the routing cubes in each layer are considered as possible location for evaluating the  $\mu$ SEL effect.

```

// Poli_VLSI
1: Read Physical Design Description of the Netlist
2: Compute Placement
3: Compute Router
4: Print GDS II
// Layer Mesh_Map
5: Read GDS II
6: FOR GDS II
7:   COMPUTE Layer_meshes
8: ENDFOR
9: FOR each Layer_mesh
10:   CALL Routing
11:   REPORT Layer_Routing_Architecture
12: ENDFOR
13: FOR each point of Layer_Routing_Arch
14:   FOR Layer=1:m
15:     IF point of Layer_Routing_Arch is in mutual layers
16:       UPDATE Latch-up Potential Report
17:     ENDIF
18:   ENDFOR
19:   PRINT Latch-up Potential Report
20: ENDFOR
21: FUNCTION Routing:
22:   Determine position of the device
23:   CASE position
24:     Selected Position in layer_mesh_1:
25:     layer_mesh_1_position==1;
26:     Selected position in layer_mesh_2:
27:     layer_mesh_2_position==1;
28:     ...
29:     Selected position in layer_mesh_m:
30:     layer_mesh_m_position==1;
31:   ENDCASE
32:   PRINT Layer_Routing_Arch

```

Fig. 6.5 A Pseudo-code overview of the developed latch-up analysis environment

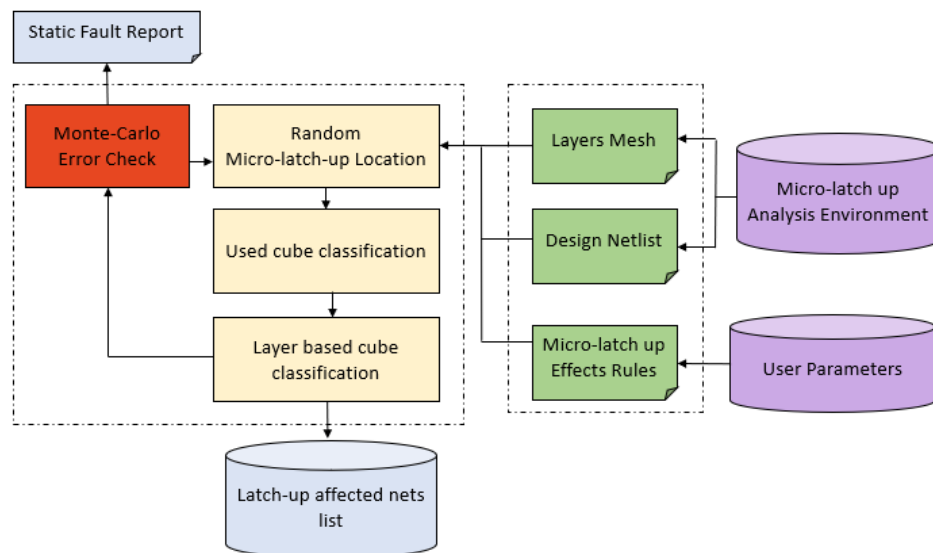


Fig. 6.6 The flow of the developed Monte Carlo Error Rate Analysis

Monte Carlo starts with reading the 3D geometry of the layout of the design and creates the layer mapping. Then, the analysis procedure is divided into several steps. Firstly, the latch up generation chooses the position for evaluating the  $\mu$ SEL effect, randomly. Moreover, in all the layers, this selected point is considered. For this random selection, all the nodes of the design are volunteered and considered. The algorithm is classifying the selected position. First of all, the algorithm controls if the selected node is considered as the used area of the device. Second aspect is related to the mutual point between routing interconnections in different layers. The chosen node will be considered as a successful possible error node only if it is a mutual node between different routing segments in different layers. This successful point is considered as the possible location of  $\mu$ SEL. error even though it is not determined.

The file containing the information related to the Netlist provides the data related to the nets containing the selected point. If the chosen point meets the requirement of the algorithm, the selected node is filtered based on the distance of the layers that the node belongs to. With respect to the chosen threshold, which is related to the design functionality, the selected nodes will be classified again.

Finally, if the chosen node does not meet the requirement of the  $\mu$ SEL effect, the algorithm comes back to the initial step. Otherwise, if the chosen location is considered as a possible  $\mu$ SEL location, the coordinates of the chosen point and also the number of Monte Carlo runs reported in the Micro Latch up Error file and the average number of runs has been updated at the end of each distribution. This procedure continues until the difference between each average is lower than the Monte Carlo Error defined by user as an effective rule and the result is reported as the static fault report.

Moreover, the Monte Carlo algorithm generated a list of the nets which are probable to generate  $\mu$ SEL in a case of radiation incident. The list of nets has been used for performing the dynamic evaluation of the design through  $\mu$ SEL Error.

### **Error Rate Generation based on Simulation Environment**

For generating Error Rate report, we generate a simulation environment that imports the affected net list generated by Monte Carlo algorithm, applying to a test bench to provide input stimuli and monitor the outputs to detect possible errors. The simula-



tion environment is generating the Error Rate report automatically by calculating the probability of detection error in the output when the reported nets are considered as a  $\mu$ SEL position. As a result, the generated Error Rate report is providing an insight of possible system behavior when the device is deployed in a radiation environment and generates the  $\mu$ SEL error.

### 6.1.2 Experimental Results

In order to confirm the accuracy and efficiency of the developed algorithm, a benchmark circuit from ITC benchmark collection has been chosen [22]. WE choose NaNGate 15nm Open Cell library used in combination with Synopsys Simplify PRO and we used the back-end NaNGate cell description in order to implement the layout.

#### Experimental Setup

The goal of the proposed environment is to choose layout with different routing congestion characteristics and test its impact on the  $\mu$ SEL sensitivity. Considering the area of the device, an area equal to  $100\mu\text{m} \times 50\mu\text{m}$ . To provide the test bench for the benchmark circuit in the simulation environment including the input stimuli, the benchmark circuit is applied to Synopsys TetraMax tool to generated the test pattern using ATPG to generate the test bench automatically. Afterwards, the logic for monitoring the output with respect the golden output is added to the test bench in order to detect errors. Finally, the faulty nets list generated from Monte Carlo algorithm, post layout design netlist and the test bench were loaded in to the simulation environment for computing Error Rate.

#### Monte Carlo Error Rate

For the selected benchmark circuit, six different routing architecture have been defined. Each architecture has been generated by changing the routing congestion, as shown in Table 6.1.

For each presented routing architecture, we start from the netlist of the design, placement and routing have been performed which at the end provide the GDS II format of the netlist. The map of each layer has been creating using GDS II format

Table 6.1 Routing Architecture Characteristic

B14 Routing Architecture	Average Routing Congestion[%]
A	33.72
B	32.97
C	34.88
D	41.63
E	52.56
F	62.11

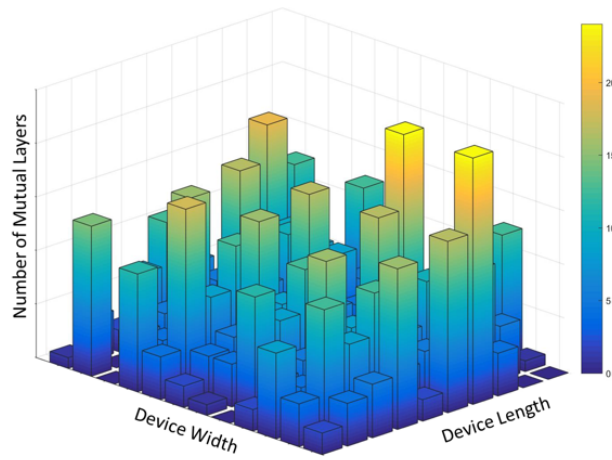


Fig. 6.7 Mutual layer distribution in terms of are width and length for benchmark B14 version F

of the netlist under the study. For each layer, the used area of the device has been extracted from the layer map of the design. The process continues by investigation of the layer distribution. Considering B14 as a selected testbench with routing architecture type F, the mutual layers occurrences are reported with respect to the device width and length is represented in Figure 6.7.

As it is shown in Figure 6.7, the distribution of the used area of the device regarding different layers is reported while the number of layers with a mutual point is defined. This information is required to identify the sensitive point of the design since the position is mutual between more layers is more sensitive to the  $\mu$ SEL fault. Please notice that the most sensitive are of the layout can be referred to the identified spot observable directly on the metal layer 1 and 2, as shown in Figure 6.8 a for the B14 benchmark, routing architecture type F. These location are drastically less

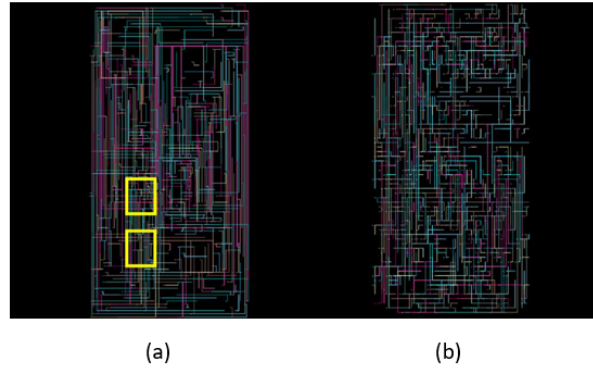


Fig. 6.8 Metal Layers 1 and 2 for the B14 implementation with routing congestion F(a) and routing congestion A(b)

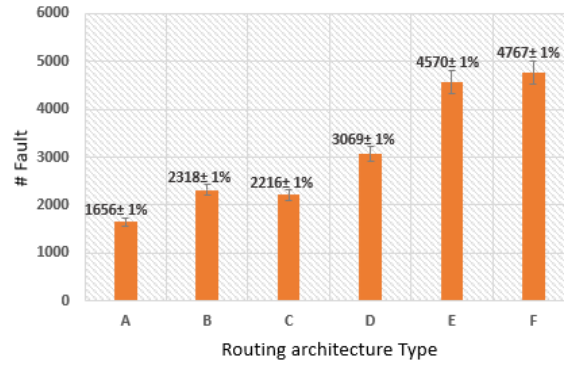


Fig. 6.9 Monte Carlo fault report for the different B14 Physical placement and routing with statistical error rate bar at 1%

congested. Therefore, less prone to  $\mu$ SEL phenomena than the correspondent one for B14 benchmark for version A, as reported in Figure 6.8, b.

Monte Carlo algorithm use the layer map of the design to estimate the average number of  $\mu$ SEL error and calculate the error rate. We evaluate the mean number of  $\mu$ SEL error fixing the tolerated precision Error 1%. Figure 6.9 is reporting the number of Monte Carlo run to reach the defined Monte Carlo error.

### Dynamic Error Rate: Fault Simulation

For evaluating the  $\mu$ SEL effect on the circuit functionality, we design a set of fault simulation experiments. For doing this, the list of the affected nets which is generated from Monte Carlo algorithm is provided as an input of the simulation environment.

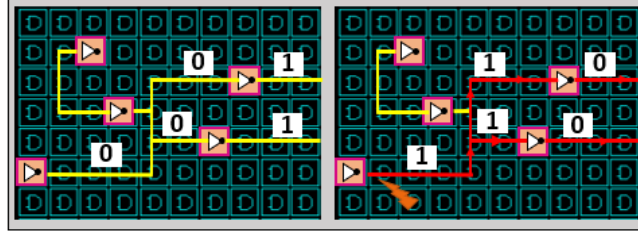
Fig. 6.10  $\mu$ SEL fault simulation instrumentation method

Table 6.2 Dynamic error rate report

Circuit	Faulty Simulation[#]	Observed Errors[#]	Error Rate[%]
A	1656	1622	97.94
B	2318	2242	96.72
C	2261	2201	97.346
D	3069	2981	97.132
E	4570	4446	97.286
F	4767	4624	97

The affected list of the nets provides the information related to the event which has been involved in the  $\mu$ SEL, successfully. Please note that the fault simulation tool is not able to directly simulate the  $\mu$ SEL effect. Therefore, we developed a simulation tool considering the propagation scenario, represented in Figure 6.10. To elaborate more, when a sensitive node is affected by a  $\mu$ SEL, it leads the propagation point of the routing branches to be simultaneously stuck at the positive logic value.

The developed simulation environment use the faulty net list with the layout netlist of the target design in order to generate the Error-rate Error. A testbench is used to apply the input stimuli to the target design, monitoring the output to compare with the golden output and record the occurrence of errors. Error rate is generated in terms of numbers of simulation runs where errors in the output have been detected, as reported in Table 6.2. The results confirm that the criticality of the  $\mu$ SEL phenomena, since almost all the single  $\mu$ SEL leads to the whole circuit misbehavior.

### 6.1.3 Research advancement on Micro Single Event Latch-up

Ultra-scale devices based on technologies below 20nm are nowadays widely adopted due to their elevated computing features and low power consumption. Using these

devices, one of the main challenges is the protection against the micro latch-up effect. Therefore, a new analysis tool for detecting the occurrence of micro latch-up event considering the physical layout of a circuit has been proposed. In details, a circuit layers has been developed in order to identify the micro latch-up sensitive points in the 3D layout geometry, while a Monte-Carlo approach has been developed to calculate the micro latch-up error rate on routing interconnection nodes. Experimental results have been performed by fault simulation on a benchmark circuit implemented in six different variants of routing congestion using a 15 nm COTS technology library demonstrating the feasibility of the proposed approach. As a next step, radiation test is planned in order to confirm the occurrence of the micro single event latch-up phenomenon. and analyzing the traditional mitigation approach and compare the results with different nano-metric technologies.

# Chapter 7

## Total Ionizing Does

Different from Single Event Effects, Total Ionizing Dose(TID) is the effect of the accumulation of the charge imposed by secondary particles interacting within the device. The amount of the accumulated charge depends on the exposure time, the flux of the particles and their LET. TID can cause several misbehavior of the system. Firstly, it cause a global worsening of the device since, it is a trapped effect in the silicon oxide of the transistor which cause the transistors to slow down and increases the power consumption of the device. Secondly, TID can increase the sensitivity of the system regarding SEU [3]. To elaborate more, the accumulated charge and the displacement damage within the crystal lattice of the device could make the device more sensitive to Single Events. Sometimes, the device can recover from TID effect rapidly. On the other side, at some cases it takes months for TID to anneal. However, one common method for annealing TID effect is heating the device. Heating provides enough energy to the crystalline lattice so that atomic locations can be restored and trapped charges can be released.

Considering recent technology such as Flash-based FPGAs, even though these devices are attracting interest fur to their high flexibility, high computing power and low power consumption, they are still subjected to cumulative TID effect, especially when they are used in mission critical applications [101] [102]. Therefore, in order to reach a successful mission, it is mandatory to evaluate the operation of our system while it is affected by TID and mitigate the system at early stage of systems development.

Several studies are dedicated to evaluate the TID effect, focusing on Flash-based FPGA devices. In [103][104], a TID experiment has been performed by exposing the Microsemi ProASIC3 Flash-based FPGA with two design under gamma rays. The first design is a chain of 2000 inverters while the second one is a chain of 2000 shift registers. As a result, they observed a degradation of 10% in propagation delay. In [105], the characterization of TID effect on Microsemi PROASIC3 Flash-based FPGAs has been performed. For this characterization, X-Ray and Gamma rays have been imposed to the chains of designs included chains of inverters and shift registers. In [2] the inverters and shift registers have been replaced with other kind of logic gates. As a result, it has been reported that the propagation delay degradation is dependent to the type of implemented logic gates.

In [106], the result of TID effect analysis of Flash-based FPGA are reported which illustrates for each component used in the system its accumulated dose before failure with comparison between design with and without reconfiguration. Recent Radiation-hardened Technology, RTG4 Flash-based FPGA, have been analyzed regarding TID effect and the result are published at [107] [108]. All the mentioned studies, are based on physical way such as accelerated particle beam and X-ray to induce TID to the device under the test and analyze the overall performance degradation of the device. However, all these approaches are sharing a common disadvantage which is: firstly, the cost of TID analysis experiment in terms of money and time. Secondly, the granularity of the analysis results is not optimal for fine optimization of the design with reliability as an important constraint.

Therefore, we propose a new work-flow for analyzing the TID effect on Flash-based FPGA, taking into account different types of gates as reported in [2] and generating an error rate reports of the target design automatically at the early stage of the development. Moreover, the environment is also evaluating the performance degradation as a result of TID effect not only in programmable logic cores but also in the routing resources.

## 7.1 The developed environment

We proposed a new environment for analyzing the TID effect on Flash-based FPGA considering the different impact of factor when the configurable logic is programmed to implement different logics in the design. The developed environment is divided in

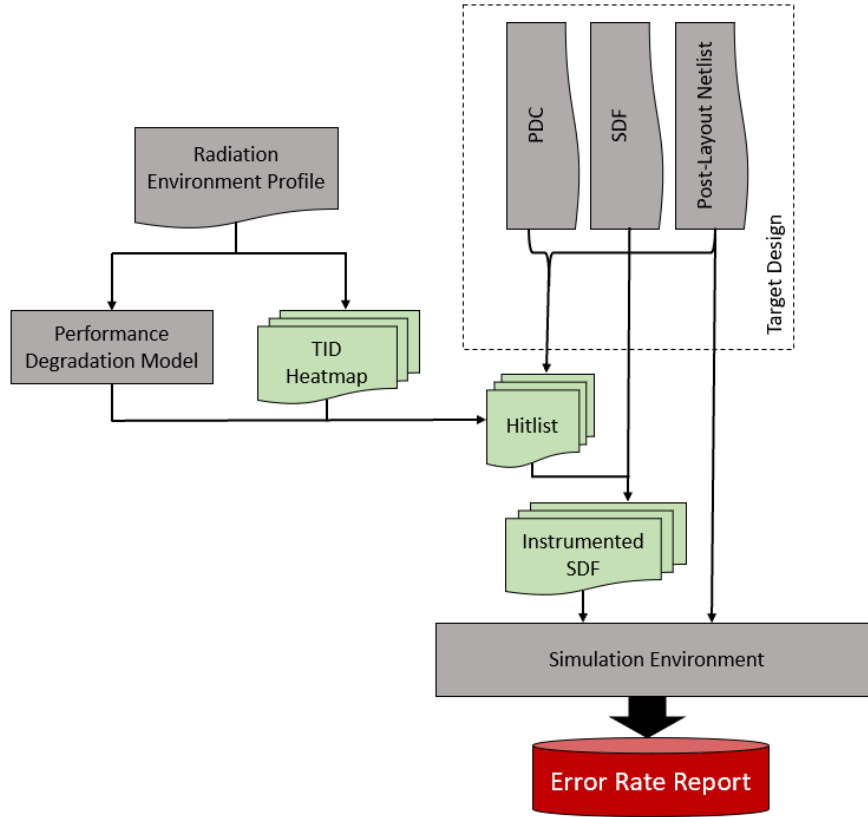


Fig. 7.1 The flow of the developed TID analysis environment

four phases: Firstly, a set of TID heatmaps is generated with respect to the radiation environment profile which is describing possible TID effect distribution. Secondly, we export information from the implemented design, including Physical Design Constraints(PDCs) file, the Simulation Delay File(SDF) and Post-layout netlist. The workflow continues with generation of Hitlist files describing the performance degradation with respect to the logic gates in the target design. Finally, the SDF file and Post-layout netlist are provided to simulation environment in order to produce error rate report for designer to gain an early stage reliability estimation of the target design regarding TID effects. Figure 7.1 represents the developed environment.

### 7.1.1 Background on Versatile architecture

Since the proposed work-flow is dedicated to Flash-based FPGAs with Versatile as a programmable logic core [1], it is beneficial to overview this architecture before



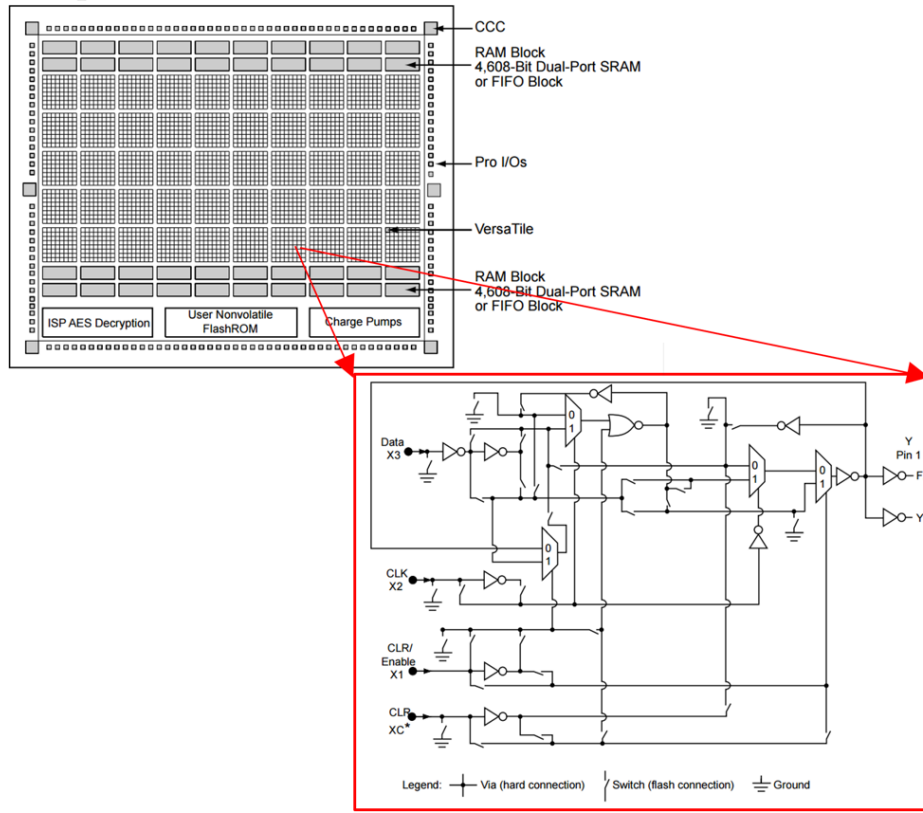


Fig. 7.2 VersaTile in Microsemi ProASIC Flash-based FPGA [1]

elaborating the proposed work-flow. Figure 7.2 represent the programmable logic core of Flash-based FPGAs.

Different circuits in the VersaTile can be generated by implementing different logic functions and changing the on/off state of the switches. Considering different number of involved switched for different logic gates, the performance degradation caused by the same TID could be different in terms of propagation delay increment. This is performed as *Performance Degradation Model(PDModel)* in our proposed environment.

### 7.1.2 Total Ionizing Dose (TID) Heatmap generation

Total Ionizing Dose(TID) effect is dependent to the amount of radiation accumulation in the whole device. Although, because of the variation of the silicon parameters,

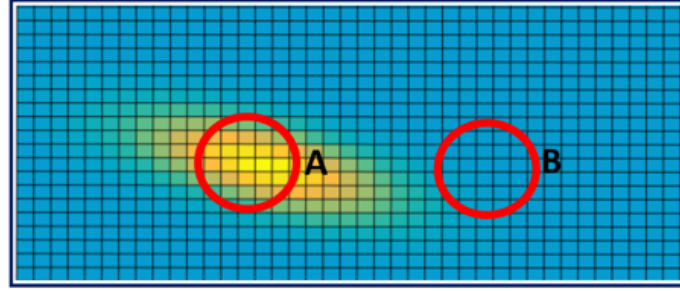


Fig. 7.3 Heatmap generation considering TID distribution: The area A represent a high density TID region, while area B is representing a TID not affecting region.

the accumulated radiation in the device is not homogeneous. Thus, we generate TID Heatmap in order to model the distribution of localized TID effect. This model has been developed in Matlab environment with the goal to provide the correlation between the radiation incident and TID effect within the device. In this modeling, a Gaussian model is used in order to estimate the particle distribution while interacting within the device. This model is receiving the dimension of the device, the radiation profile and the total dose determined by the radiation profile. Then, it selects randomly a hotspot location in the device which is defined as the location that charges are expected to accumulate more densely. Figure 7.3 as an example of the device Heatmap evaluating the TID distribution, illustrates that area A of the device as an affected cell suffers from the performance degradation of the cell while cell located in area B of the device work at the original speed.

The scheme of the developed work-flow is represented in Figure 7.4. The algorithm goes on with the elaborating the characteristics of the device under the study such as device size and generates a simplified FPGA map. It continues with selecting several hotspot as sensitive points of the device. For each selected hotspot, the TID effect is modeled considering the progressive energy accumulation induced by the particles, whose profile is specified by two parameters, namely Particle Quantity and Particle Energy. Moreover, we define a dynamic TID coefficient to map the discharged energy of the particle to the absorbed energy by the devices and eventually the TID considering the Gaussian particles since the TID is accumulated non-linearly. Likewise, the algorithm sets intermediate observation point while TID has been accumulated to the final target dose specified by the user in order to provide a better granularity for analyzing the performance.

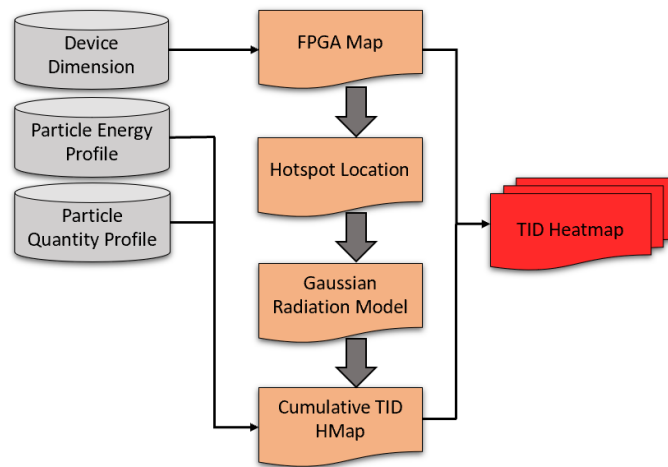


Fig. 7.4 Block diagram of TIC Heatmap generation

Along with the Heatmap, depend on the radiation environment profile and target FPGA device profile, a PDModel is developed. The PDModel consists in: Firstly, the propagation delay model within the VersaTile considering the logic implemented in the design. Secondly, a propagation delay model for the routing resources connecting the logic functions among different VersaTiles used in the design. These two models are adopted as increment of path delay inside the gates and port to port delay among different gates respectively during the phases of Hitlist generation and SDF instrumentation.

### 7.1.3 Hitlist generation

With respect to the PDModel and generated TID heatmaps, a set of Hitlists is generated which is describing the performance degradation caused by TID effects in each cell of the design. The generated Hitlist files follow the mechanism used in the SDF to annotate timing information of the target design. Therefore, our developed environment is receiving the original SDF generated by Mircosemi Libero tools uses IOPATH directive to annotate the delay from input port to output port inside a cell and PORT directive to annotate propagation delay of the net from port of a cell to port of another cell. Therefore, the Hitlis file adopt the similar names to mark the delay increment of affected gates and net paths. Figure 7.5 is representing the pseudo code of the developed algorithm for generating the Hitlist.

---

```

1: pdc_cells= all the instance in design //cell in pdc_file
2: hmfiles= 1000 heat maps
3: PDM= performance degradation model
4: for each heatmap in hmfiles:
5:     hitlist=new Hitlist(sdf_origin)
6:     for each cell in pdc_cells:
7:         if isCellIrrated(heatmap, cell):
8:             delayInc=calCellDelayInc(heatmap, cell, PDM)
9:             hitlist.write(delayInc)
10: hitlist.close()

```

---

Fig. 7.5 Algorithm for generation Hitlist

The developed tool firstly import the information from PDC file which contains the placement of cell instances programmed and used in the design. The PDC file includes characteristics of the placement of cell instances programmed and used in the design. For each TID heatmap, the tool generated a hitlist considering two conditions: Firstly, the VersaTile of the placed cell is affected by TID effects directly which leads to delay increment inside cell. Secondly, the path connected to the input ports of the cell is affected which leads to delay increment of the interconnection.

For the first condition, the tool is developed in a way that takes into account an important features which is considering also the architecture of the VersaTiles. Therefore, when the VersaTile is programmed to implement different types of logic gates, the performance degradation is different with same TID. Therefore, the tool elaborate the information of the cell placed(programmed) in the affected VersaTile, such as type of the cell, used input and output port and merge them with the information in heatmap and the PDModel to calculate delay increment within the cell.

Differently, for the second condition, it is essential to extract the routing information from the design with the FPGA used commercial design tool. Therefore, we build a simplified hypothetical routing model with the following assumption:

1. Presence of just horizontal or vertical connection between the source port and destination port.
2. Evenly distribution of propagation delay along the path.

---

```

1: hitlists= a set of hitlist generated previously
2: for each hitlist in hitlists:
3:     instrumentedSDF= new SDF(sdf_origin)
4:     for each delayInc of hitlist:
5:         originalDelayVal= findDelayVal(sdf_origin, delayInc,targetCell, delayInc,type)
6:         incDelayVal=calDelayInc(originalDelayVal, delayInc,inc)
7:         instrumentedSDF.write(delayInc.targetCell,incDelayVal)
8:     instrumentedSDF.close()

```

---

Fig. 7.6 Algorithm for generation of instrumented SDF.

With the placement information of the source port and destination port, the heatmap and PDMModel, the delay increment of the affected path is calculated using the equation 7.1 in which  $Seg_i$  is each segment of routing along the target path,  $Delay(PATH)$  is the path delay extracted from original SDF and  $\partial$  is the performance degradation co-efficient determined by the PDMModel with respect to delay increment.

$$PathDelayIncr = \sum_{seg_i} \frac{Length(Seg_i)}{Length(PATH)} * Delay(PATH) * \partial \quad (7.1)$$

### 7.1.4 SDF Instrumentation

The delay information extracted from the original SDF during the back-annotation phase of standard FPGA design flow and the generated Hitlist, a collection of instrumented SDF is generated reflecting different possible distributions of performance degradation across the target device taking into account of the place and layout information of the target design extracted from the Physical Design Constraint(PDC) file. Figure 7.6 is representing the algorithm for generation the instrumented SDF from Hitlist file.

### 7.1.5 Simulation Execution

In order to estimate the error rate information, we developed a simulation environment that uses instrumented SDFs and Post-layout of the target design as inputs. Then, it uses a test-bench to provide input stimuli, which can be external test pattern of the circuit or internal patterns and monitor the outputs to detect possible error.

The simulation environment is calculating the probability of detection error in the output considering a specific accumulated amount of TID and generate the error rate

report. Moreover, the information regarding the output where the error is detected, the site and time of timing-check violation is reported by the simulation environment. This information is beneficial for the designer to analyze the behavior of the design considering the performance degradation introduced by TID and apply a possible mitigation technique.

## **7.2 Experimental Results**

In order to confirm the feasibility of the proposed work-flow, we choose five circuits from TIC99 benchmark collection [22]. The characteristics of the chosen circuits are reported in Table 7.1. The commercial design flow were used to generate the Post-layout netlist, the PDC file and SDF back annotation file, which were later used as input of the proposed flow.

Table 7.1 Benchmark circuits characteristics

Benchmark	Period[ns]	Core Utilization in FPGA#	Core Utilization in FPGA %	FF[#]	PI[#]	PO[#]
b11	10	385	6.26	34	7	6
b12	11	553	9.00	123	5	6
b13	5.5	133	2.16	56	10	10
b14	34	3698	60.19	216	32	54
b15	23	4569	74.37	435	36	70

### 7.2.1 Experimental Setup

The chosen circuits are implemented on ProASIC3 A3P250 Flash-based FPGA with a total of 6144 VersaTiles. Since satellites and space probes typically encounter TID between 10 and 100 krad(Si) [109], for each bench-mark circuits 250 Heatmaps were generated. In order to do this, the assumption was that particles hitting a VersaTile generated 50 krad(Si) radiation, considering that the nearby VersaTiles will also suffer performance degradation following Gaussian distribution. As it has been mentioned before, we set 4 intermediate observation points for accumulation of TID equal to 25%, 50%, 75% and 100%. 100% has been used to generate the reports with better granularity for analyzing the performance degradation along the mission life-time.

Two main models were applied in the PDMModel for gate propagation delay and net propagation delay. Regarding the first model, data from [2] was used for different performance degradation factor when the VersaTile is programmed to implement various logic functions. Due to different implemented logic functions, the path from input port to output port inside the gate is different. For the same dose, the performance degradation in terms of delay increment is different with respect to the gate type, represented in Figure 7.7. Results reported in [2] for different types of gates are used in our model. However, we assume that the performance degradation is equivalent for the gates in the chain.

Following, the original delay values with respect to the Manhattan distance between two ports of gates placed in two different VersaTiles were extracted as it is shown in Figure 7.8. The data was extracted by:

1. Create a circuit design with two inverters.
2. Set the placement constraints to make the two inverters have desired horizontal and vertical distance.
3. Synthesis and implement the design on the target FPGA.
4. Export the back-annotation SDF during Post-layout phase.
5. Extract the delay value from the SDF.
6. Report above steps for every combination of horizontal and vertical distances.



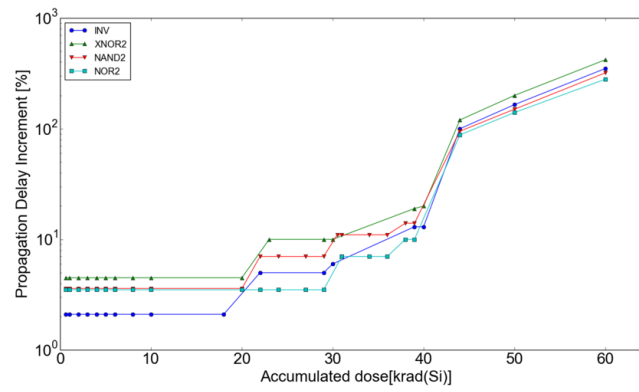


Fig. 7.7 Performance degradation for different types of gates [2]

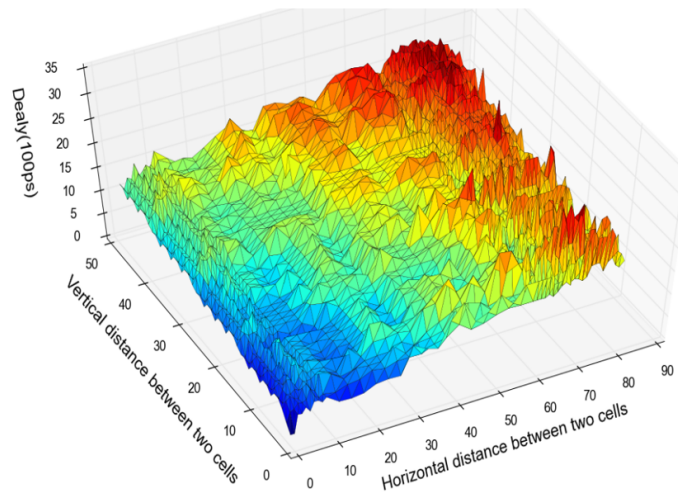


Fig. 7.8 Net propagation delay with respect to Manhattan distance between net source and destination(X distance and Y distance separated)

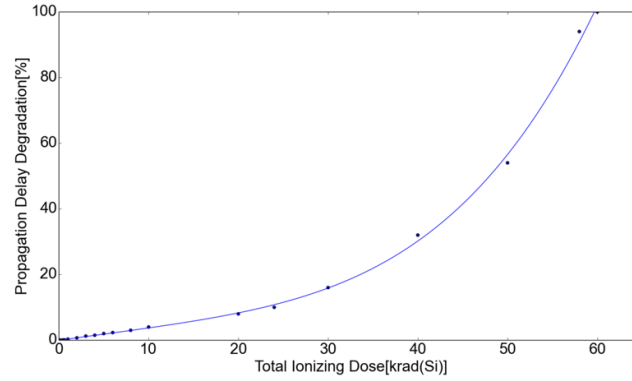


Fig. 7.9 Performance degradation coefficient model for routing net

A simple performance degradation model of the routing resources is build based on the experiment reported on [108] which has been described in the Equation 7.1 and represented in Figure 7.9.

Adopting the placement information for locating the list of the cells, Heatmap and Hitlist for each bench-mark were generated. The Hitlist contains the degradation for each affected cell in terms of delay increment and then was used to instrumented SDF. In generation of Heatmap in our developed environment and the selected benchmark placement, not all the logic functions are exposed to TID effect.

The selected bench-mark circuits were implemented using Synopsys TetraMax tool to generate test pattern using ATPS to automatically produce test-bench. Then, the logic for monitoring the output and compare it with the golden output to detect the errors was also added to the test-bench.

Finally, the instrumented SDF, Post-layout design netlist and the test-bench were loaded into the simulation environment for automatically generation of error rate report.

### 7.2.2 Error Rate Reports

Our proposed technique has been used to calculate the error rate of the bench-mark circuits. To elaborate more, different error rates are calculated in four different stages of accumulated dose effect. Figure 7.10 represent the calculated error rate.

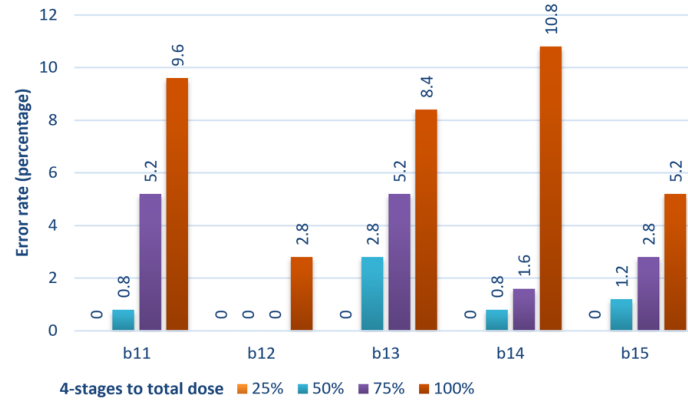


Fig. 7.10 Error rate results of the selected ITC99 bench-mark with respect to different percentage for TID

AS it can be noticed from the Figure 7.10, the error rate increase with respect to TID accumulation. However, the attractive results are related to the time or dose in which the error is observed in the output. This result together with the information of the location and time of the error and timing-check fail, are essential for a designer to

As a last step, we performed the experiment with the heatmap where same energy was distributed evenly across the chip. Comparing the result of this experiment with the one explained before with the nonlinear distributions of TID effect, the results shows that the even distribution leads to underestimation of error rate for the target design which is not preferable during long term space mission planning.

### 7.2.3 Research advancement on Total Ionizing Does

In this paper, we presented a new workflow for analyzing TID effects on Flash-based FPGA taking advantage of instrumenting the simulation delay data from back-annotation phase of commercial FPGA design flow. With the proposed workflow, it is possible to model the performance degradation in terms of delay of both the propagation delay inside the cell and the interconnection delay of route resources induced by TID effects. Five benchmark circuits were used to demonstrate the feasibility for automatically generating error rate report, which can be used as early stage system assessment regarding TID effects. Considering the fact that the Heatmap generation and PDMModel used for Hitlist generation and s instrumentation are in the

primary stage, radiation test experiments are outlined to provide more information for refining the model and validating the precision of the proposed workflow.

# References

- [1] Microsemi, proasic3 fpga fabric use guide.
- [2] R. Vaz F. Kastensmidt, E. Fonseca. Tid in flash-based fpga: Power supply-current rise and logic function mapping effects in propagation-delay degradation. *IEEE Transactions on Nuclear Science*, 58:1927–2934, 2011.
- [3] B. Battezzati M. Violante, L. Sterpone. *Reconfigurable Field Programmable Gate Array for Mission-Critical Applications*. Springer, 2011.
- [4]
- [5]
- [6] R. Baumann. Radiation-induced soft errors in advanced semiconductor technologies. 2005.
- [7] J. Tausch D. Sleeter. Neutron induced micro sel events in cots sram devices. 2007.
- [8] A. Dixit A. Wood. The impact of new technology on soft error rates. 2011.
- [9] P. Dodd. Physics-based simulation of single-event effects. *IEEE Trans. Device Mater. Reliab*, 5:343–357, 2005.
- [10] M. Gadlage J. Benedetto, P. Eaton. Digital device error rate trends in advanced cmos technologies. *IEEE Transaction on Device and Materials Reliability*, 53:3466–3471, 2006.
- [11] F.L. Kastensmidt L. Sterpone, N. Battezzati. An analytical model of the propagation induced pulse broadening (pipb) effects on single event transient in flash-based fpgas. *IEEE Transacriion on Nuclear Science*, 58:2333–2340, 2011.
- [12] M. Violante F. Abate, L. Sterpone. A study of the single even effects impact on functional mapping within flash-based fpgas. 2009.
- [13] V. Ferlet L. Sterpone, N. Battezzati. Analysis of set propagation in flash-based fpgas by means of electrical pulse injection. *IEEE Transacriion on Nuclear Science*, 57:1820–1826, 2010.

- [14] Z. Huang H. Liang, X. Xu. A methodology for characterization of set propagation in sram-based fpgas. *IEEE Transacriion on Nuclear Science*, 63:2985–2992, 2016.
- [15] M. Benedetto D. Mavis, M. Gadlage. Digital single event transient trends with technology node scaling. *IEEE Transacriion on Nuclear Science*, 43:3462–3465, 2006.
- [16] J.J. Wang N. Rezzak, D. Dsilva. Set and sefi characterization of 65 nm smartfusion2 flash-based fpga under heavy ion irradiation. 2015.
- [17] D. McMorro V.F.Cavrois, V. Pouget. Investigation of the propagation induced pulse broadening (pipb) effect on single event transients in soi and bulk inverter chains. *IEEE Transacriion on Nuclear Science*, 55:2842–2853, 2008.
- [18] V. F. Cavrois L. Sterpone, N. Battezzati. Analysis of set propagation in flash-based fpgas by means of electrical pulse injection. *IEEE Transacriion on Nuclear Science*, 57:1820–1826, 2010.
- [19] J. M. Benedetto D. G. Mavis, M. Gadlage. Digital single event transient trends with technology node scaling. *IEEE Transacriion on Nuclear Science*, 43:3462–3465, 2006.
- [20] H. S. Chen J. J. Wang, S. Samiee. Total ionizing does effects on flash-based fpgafield programmable gate array. *IEEE Transacriion on Nuclear Science*, 51:3759–3766, 2004.
- [21] A. Manuzzato N. Battezzati, S. Gerardin. Methodologies to study frequency-dependent single event transient effects sensitivity in flash-based fpgas. *IEEE Transacriion on Nuclear Science*, 56:3534–3541, 2009.
- [22] G. Squillero F. Corno, M. Reorda. Rt-level itc99 benchmarks and first atpg results. *IEEE Design and Test of Computers*, 17:44–53, 2000.
- [23] M. Ebrahimi R. Bishnoi, H. Asadi. Layout-based modeling and mitigation of multipe even transients. *IEEE Transaction on Computer-Aided Desing of Intergrated Circuits and Systems*, 35:367–379, 2015.
- [24] L. Sterpone S. Azimi, B. Du. On the prediction of radiation-induced sets in flash-based fpgas. *Elsevier Microelectronic Reliability*, 64:230–234, 2015.
- [25] L. Sterpone S. Azimi, B. Du. Radiation-induced single event transients modeling and testing on nanometric flash-based techniques. *Elsevier Microelectronic Reliability*, 55:2087–2019, 2015.
- [26] F. Ribeiro G. Wirth, F. Kastensmidt. Single event transient in logic circuits-load and propagation induced pulse broadening. *IEEE Transacriion on Nuclear Science*, 58:2928–2935, 2008.
- [27] A. Wood A. Dixit. The impact of new technology on soft error rates. 2011.

- [28] D. . Mavis M. Benedetto, P. Eaton. Digital single event tranient trends with technology node scaling. *IEEE Transaction on Nuclear Science*, 53:3462–3465, 2006.
- [29] Gerardin N. Battezzatim, S. Methodologies to study frequency dependent single event transient effects sensitivity in flash-based fpgas. *IEEE Transaction on Nuclear Science*, 56:3534–33541, 2009.
- [30] S. Azimi L. Sterpone. Radiation-induced set on flash-based fpgas: Analysis and filterng methods. 2017.
- [31] A. Manuzzato N. Battezzati, S. Gerardin. On the evaluation of radiation-induced transient faults in flash-based fpgas. 2008.
- [32] E. Ting S. Rezgui, J. Wang. New methodologies for set characterization and mitigation in flash-based fpgas. *IEEE Transaction on Nuclear Science*, 54:2512–2524, 2007.
- [33] S. Azimi L. Sterpone, B. Du. Radiation-induced single event transients modeling and testing on nanometric flash-based technologies. *Microelectronic Reliability*, 55:2087–2091, 2015.
- [34] M. Valdersa M. Garbayo, M. Garcia. C-element model for set fault emulation. 2013.
- [35] Microsemi. aa. Using synplify to desing in microsemi radiation-hardened fpgas2. 2012.
- [36] L. Sterpone D. Sabena, M. Reorda. On the evaluation of soft-errors detection tehcniques for gpgpus. 2013.
- [37] [http://www.esa.int/Our Activities/Space-Science/COROT](http://www.esa.int/Our_Activities/Space-Science/COROT). Esa corot missin documentation. 2014.
- [38] <http://www.across-project.eu/workshop2013/121108> ARAMIS-Introduction-HiPEAC-WS-V3.pdf. Aramis project overview. 2013.
- [39] M. Reorda. L. Sterpone D. Sabena. evaluating the radiation sensitivity pf gpgpu caches: New algorithms and experimental results. *Elsevier Microelec-tronic Reliability*, 54:2621–2628, 2014.
- [40] L. Carro L. Gomez, F. Capello. Gpgpus: how to combine high computational power with high reliability. 2014.
- [41] C. Frost P. Rech, C. Aguiar. An efficient and experimentally tuned software-based hardening strategy for matrix multiplication on gpus. *IEEE Transaction on Nuclear Science*, 60:2797–2804, 2013.
- [42] L. Carro d. Sabena, L. Sterpone. Reliability evaluation of embedded gpgpus for safety critical applications. *IEEE Transaction on Nuclear Science*, 61:3123–3129, 2014.

- [43] L. Sterpone D. Sabena, M. Reorda. On the evaluation of soft-errors detection techniques for gpgpus. 2013.
- [44] T. Li J. Tan, N. Goswami. Analyzing soft-error vulnerability on gpgpu microarchitecture. 2011.
- [45] K. Pattabiraman B. Fang, J. Wei. Towards building error resilient gpgpu applications. 2012.
- [46] D. Defour S. Collange, M. Daumas. A parallel functional simulator for gpgpu. 2010.
- [47] W. Fung A. Bakhoda, G. Yuan. Analyzing cuda workloads using a detailed gpu simulator. 2009.
- [48] G. Yuan W. Fung, I. Sham. Dynamic warp forming and scheduling for efficient gpu control flow. 2014.
- [49] <http://www/gpgpu-sim.org/>.
- [50] M. Ripeanu B. Fang, K. Pattabiraman. A methodology for evaluating the error resilience of gpgpu applications. 2014.
- [51] S. Halder H. Wunderlich, C. Braun. Efficacy and efficiency of algorithm-based fault tolerance on gpus. 2013.
- [52] L. Sterpone S. Azimi, B. Du. Radiation-induced single event transients modeling and testing on nanometric flash-based technology. *Elsevier Micro-electronic Reliability*, 55:2087–2091, 2015.
- [53] J. Abraham K. Huang. Algorithm-based fault tolerance for matrix operations. *IEEE Transaction on Computer*, C-33:518–528, 2009.
- [54] Nvidia cuda programming guide. 2009.
- [55] Ogloo, proasic3. 2010.
- [56] R. Tessier k. Andryc, M. Merchant. Flexgrip: A soft gpgpu for fpgas. 2013.
- [57] Microsemi aa. vv. Using synplify to design in microsemi radiation-hardened fpgas2. 2012.
- [58] D. McMorrow M. Baze, S. Buchner. A digital cmos design techniques for seu hardening. *Transactions on Nuclear Science*, 47:2603–2608, 2000.
- [59] Microsemi aa.vv. Using synplify to design in microsemi radiation-hardened fpgas2. 2012.
- [60] B. Du L. Sterpone, S. Azimi. Effective mitigation of radiation-induced single event transient on flash-based fpgas. 2017.



- [61] R. Won S. Rezgui, J. McCollum. Desing and layout effects on set propagation in 90-nm asic and fpga test structures. *IEEE Transactions on Nuclear Science*, 57:3716–3724, 2010.
- [62] E. Tung S. Rezgui, L. Wang. New methodologies for set characterization and mitigation in flash-based fpgas. *IEEE Transactions on Nuclear Science*, 54:2512–2524, 2007.
- [63] L. He Y. Lein. Device and architecture concurrent optimization for fpga transient soft error rate. 2007.
- [64] Z. Zhang C. Chen, S. Song. An fpga-based transient error simulator for resilient circuit and system desing and evaluation. *IEEE Transactions on Circuits and Systems III: Express Briefs*, 62:471–475, 2015.
- [65] B. Du L. Sterpone, S. Azimi. A selective mapper for the mitigation of sets on rad-hard rtg4 flash-based fpgas. 2016.
- [66] L. Sterpone S. Azimi, B. Du. Seta: A cad tool for single event transient analysis and mitigation on flash-based fpgas. 2018.
- [67] Y. Sun S. Rezgui, J. Wang. Configuration and routing effects on the set propagation in flash-based fpgas. *IEEE Transactions on Nuclear Science*, 55:3328–3335, 2008.
- [68] M. Violante F. Abate, L. Sterpone. A study of the single event effects impact on functional mapping within flash-based fpgas. 2009.
- [69] UG0574 Userguide. Rtg4 fpga fabric. 2015.
- [70] Microsemi aa.vv. Using synplify to design in microsemi radaiton-hardened fpgas2. 2009.
- [71] B. Du L. Sterpone. Analysis and mitigation of singele event effects on flash-based fpgas. 2014.
- [72] M. Zhang S. Mitra, N. Seifert. Robust system desing with built-in soft error resilience. *IEEE Computer Society*, 38:43–52, 2005.
- [73] Xilinx. Inc. Xilinx xtmrtool user guide, ug156(v2.1). 2006.
- [74] P. Rech J. Tonaft, L. Kastensmidt. Analyzing the effectiveness of a frame level redundancy scrubbing technique for sram-based fpgas. *IEEE Transaction on Nuclear Science*, 62:3080–3087, 2015.
- [75] A. Das R. Santos, S. Venkataraman. Criticality aware scrubbing mechanism for sram-based fpgas. 2014.
- [76] L. Sterpone B.Du, M. Desogus. Analysis and mitigation of seus in arm-based soc on xilinx virtex-v sram-based fpgas. 2015.

- [77] V. Izzo R. Giordano, S. Perrella. A redundant-configuration scrubbing of sram-based fpgas. *Transactions on Nuclear Science*, 64:2497–2504, 2017.
- [78] Xilinx. Inc. 7 series fpgas data sheet: Overview- product specification. 2017.
- [79] Digilent. Zybo reference manual.
- [80] J. Ranft A. Ferraro, A. Fasso. Fluka: A multi-particle transport code. 2005.
- [81] D. Codinachs M. Desogus, L. Sterpone. Validation of a tool for estimating the effects of soft-errors on modern sram-based fpgas. 2014.
- [82] Xilinx. Inc. 7series fpgas configuration user guide- ug470.
- [83] G. Swift D. Lee, M. Wirthlin. Single event characterization of the 28-nm xilinx kintex-7 field programmable gate array under heavy ion irradiation. 2014.
- [84] G. Swift M. Wirthlin, D. Lee. A method and case study on identifying physically adjacent multiple-cell upsets using 28-nm, interleaved and seeded-protected arrays. *Transactions on Nuclear Science*, 61:3080–3087, 2014.
- [85] D. Petrick M. Berg, C. Poivey. Effectiveness of internal versus external seu scrubbing mitigation strategies in a xilinx fpga: Design, test and analysis. *Transactions on Nuclear Science*, 55:2259–2266, 2008.
- [86] M. Vallejo I. Alzu. Design techniques for xilinx virtex fpga configuration memory scrubbers. *Transactions on Nuclear Science*, 60:376–385, 2013.
- [87] R.Boberg A. Tylka, J. Adams. Creme96: A revision of the cosmic ray effects on micro-electronics code. *IEEE Transactions on Nuclear Science*, 44:2150–2160, 2010.
- [88] M. R. Shaneyfelt J. R. Schwan, V. F. Cavrois. Radiation effects in soi technologies. *IEEE Transactions on Nuclear Science*, 50:522–538, 2003.
- [89] R. R. Troutman. Latchup in cmos technologies. *IEEE Circuits and Devices Magazine*, 3:15–21, 1987.
- [90] D. Radaelli J. Tausch, D. Sleeter. Neutron induced micro sel events in cots sram devices. 2007.
- [91] J. J. Wang N. Rezzak. Single event latch-up hardening using tcad simulations in 130 nm and 65 nm embedded sram in flash-based fpgas. *IEEE Transactions on Nuclear Science*, 62:1599–1600, 2015.
- [92] F. Braud D. Truyen, E. Leduc. Elimination of single event latch-up in the atmel atmx150rha rad-hard cmos 150nm cell-based asic family. 2015.
- [93] G. Hubert L. Artola, N. J. H. Roche. Analysis of angular dependence of single event latch-up sensitivity for heavy-ion irradiations of 0.18 cmos technology. *Transactions on Nuclear Science*, 62:2539–2546, 2015.

- [94] J. Y. Lee S. H. Jeong, N. H. Lee. New approach for transient radiation spice model of cmos circuit. *Journal of Electrical Engineering and Technology*, 8:1182–1187, 2013.
- [95] G. Hubert SL. Artola. Modeling of elevated temperature impact on single event transient in advanced cmos logics beyond the 950nm technology node. *IEEE Transaction on Nuclear Science*, 61:4421–4429, 2013.
- [96] L. Artola G. Hubert. Single event transient modeling in 65 nm bulk cmos technology based on multi-physical approach and electrical simulations. *IEEE Transaction on Nuclear Science*, 60:4421–4429, 2013.
- [97] T. Rousselin L. Artola, G. Hubert. Single event latchup modeling based on coupled physical and electrical transient simulations in cmos technology. *IEEE Transaction on Nuclear Science*, 61:3543–3549, 2014.
- [98] M. Ning G. Yue, W. Shaojun. A single event latch-up protection method for sram fpga. 2017.
- [99] M. Nicolaidis. *Soft Errors in Modern Electronics Systems*. Springer, New York City, 2011.
- [100] D. Radaelli J. Tausch, D. Sleeter. Neutron induced micro sel events in cots sram devices. 2007.
- [101] F. McLean T. Oldham. Total ionizing dose effects in mos oxides and devices. *IEEE Transactions on Nuclear Science*, 50:483–499, 2003.
- [102] H. Barmaby. Total ionizing dose effects in modern cmos technologies. *IEEE Transactions on Nuclear Science*, 53:3103–3121, 2006.
- [103] Y. Sun S. Rezgui, J. Wang. Tid characterization of 0.13um flash-based fpgas. 2008.
- [104] Y. Sun S. Rezgui, J. Wang. New reprogrammable and non-volatile radiation tolerant fpga: Rta3p. 2008.
- [105] <https://www.microsemi.com/document-portal/doc-view/131374-radiation-tolerant-proasic3-fpgas-radiation-effect-report>. Microsemi, radiation-tolerant proasic3 fpgas radiation effects available online. 2008.
- [106] F. Kastensmidt J. Tarrillo, J. Azambuja. Analyzing the effects of tid in an embedded system running in a flash-based fpga. *IEEE Transactions on Nuclear Science*, 58:2855–2862, 2011.
- [107] D. Dsilva N. Rezzak, J. Wang. Tid and see characterization of microsemi 4th generation radiation tolerant rtg4 flash-based fpga. 2015.
- [108] et al J. Wang. Flash-based fpga tid and long term retention reliability thorough vt shift investigation. 2015.
- [109] A. Macdonald J. Wall. The nasa asic guide assuring asics for space. 1993.

# Appendix A

## Research Achievements

The golden part of this dissertation is dedicated to Single Event Transient, covering different phases of this phenomenon, from the generation until the mitigation. In order to achieve this goal, several tools and algorithms have been developed to perform an accurate analysis, identifying the behavior of the technology regarding Single Event Transient and mitigate the design with respect to the performed analysis. This comprehensive chain of tools for analyzing and mitigation of Single Event Transient has been applied to several industrial projects such as EUCLID space mission project with the goal of monitoring the dark space which the launch is planned for 2020 carrying by European Space Agency. **The developed SET analysis and mitigation workflow has been part of the handbook Space Product Assurance Techniques for Radiation Effects Mitigation in ASICs and FPGAs handbook, published by European Space Agency.**

Moreover, I have presented the developed chain of tools in the competition organized by IEEE Council on Electronic Design Automation and the tool chain has been known as the **Best EDA Tool for improving design automation for integrated circuits and systems by IEEE Council on Electronic Design Automation granted 1000 USD.** Moreover, the developed tools have been a relevant part of several collaboration contracts with European Space Agency.

In order to confirm the reliability of the developed tool chain, I have participated in several radiation tests in collaboration with European Space Agency, CERN, Thales Alenia Space and OHB Italia, performed in several radiation chambers at CERN facility. Several different devices have been disposed to the radiation beam such

as ProASIC3 Flash-based FPGA, Xilinx SRAM-based FPGA and SMART fusion Flash-based FPGA in order to study the effect of Single Event Transient and Single Event Upset on the functionality of the device under the test.